GrowOVER: How Can LLMs Adapt to Growing Real-World Knowledge?

Dayoon Ko Jinyoung Kim Hahyeon Choi Gunhee Kim

Seoul National University

dayoon.ko@vision.snu.ac.kr jiny1623@snu.ac.kr gk0gus0@snu.ac.kr gunhee.kim@snu.ac.kr

https://github.com/dayoon-ko/GrowOVER

Abstract

In the real world, knowledge is constantly evolving, which can render existing knowledge-based datasets outdated. This unreliability highlights the critical need for continuous updates to ensure both accuracy and relevance in knowledge-intensive tasks. To address this, we propose **GrowOVER-QA** and **GrowOVER-Dialogue**, dynamic open-domain QA and dialogue benchmarks that undergo a continuous cycle of updates, keeping pace with the rapid evolution of knowledge. Our research indicates that retrieval-augmented language models (RaLMs) struggle with knowledge that has not been trained on or recently updated. Consequently, we introduce a novel retrieval-interactive language model framework, where the language model evaluates and reflects on its answers for further re-retrieval. Our exhaustive experiments demonstrate that our training-free framework significantly improves upon existing methods, performing comparably to or even surpassing continuously trained language models.

1 Introduction

In natural language research, many knowledge-intensive tasks have been actively studied, including open-domain QA (Kwiatkowski et al., 2019; Joshi et al., 2017; Yang et al., 2018), fact-checking (Thorne et al., 2018), entity linking (Hoffart et al., 2011), and open-domain dialogue (Dinan et al., 2018), to name a few (Petroni et al., 2021; Levy et al., 2017). Such a knowledge-intensive task is mostly to utilize world knowledge to generate a proper answer for a given query (Lewis et al., 2020). However, the amount of real-world knowledge is often too enormous for models to fully store them in the parameters. Thus, in most scenarios, a retriever is employed to seek relevant paragraphs or documents in a knowledge bank, such as a vector database, and a generator provides answers based on the retrieved passages or documents (Lewis et al., 2020; Guu et al., 2020). Previous benchmarks (Kwiatkowski et al., 2019; Dinan et al., 2018; Petroni et al., 2021) annotate gold answers and the evidence text needed to predict them, and evaluate the retriever using the evidence text and the generator using the gold answers.

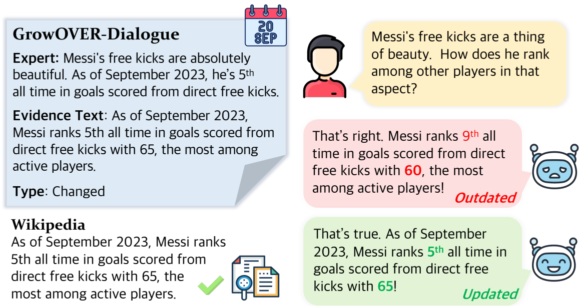

Figure 1: An illustration of GrowOVER benchmarks. GrowOVER is automatically generated and continuously updated. It provides the evidence text to evaluate the retriever and also comprehensively evaluates the generator through an open-domain dialogue task.

In the real world, new knowledge is constantly being created, and existing knowledge is changing over time, causing annotated benchmarks to become quickly outdated. To handle this issue, Kim et al. (2023) and Margatina et al. (2023) respectively suggest dynamic QA and cloze query benchmarks, which are automatically generated by comparing two Wikipedia (or Wikidata) snapshots at different times. However, they provide no annotated evidence text for the gold answers, which may make it difficult to evaluate retrievers in open-domain knowledge-intensive tasks. In the retrieval-augmented generation (RAG) framework, it is crucial to accurately measure the performance of each component as well as end-to-end performance. This enables precise identification of error sources and inaccuracies,

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3282–3308

	TempLAMA (2022)	RealtimeQA (2022)	DynamicTempLAMA (2023)	TemporalWiki (2022)	EvolvingQA (2023)	GrowOVER (Ours)
Label types	C	C	C, U, N	C, U	C, U, N	C, U, N
Automation	✗	✗	✓	✓	✓	✓
Maintenance	✗	✗	✗	✗	✗	✓
Evidence text	✗	✗	✗	✗	✗	✓
Tasks	Cloze query	QA	Cloze query	✗	QA	QA & Dialogue

Table 1: Comparison of our GrowOVER with existing benchmarks. The Label Types display the data types available in each dataset with Changed, Unchanged, and New. The automation indicates the feasibility of automatic generation. The Maintenance represents whether the validity of previously generated datasets is verified in the forthcoming time step. The evidence text indicates whether the dataset includes the evidence text. Lastly, the tasks identify the intended tasks for each dataset.

allowing for less frequent updates to Large Language Models (LLMs). Also, the tasks of these benchmarks require the model to provide only direct answers, which mainly consist of entities. In contrast, real-world knowledge can't be structured as simple question-and-answer pairs. Instead, it's more accurately represented as a vast, interconnected knowledge graph. Therefore, there are limitations in evaluating the generator's ability to provide contextually appropriate and informative answers that incorporate relevant background knowledge.

To overcome such limitations, we propose novel open-domain dynamic benchmarks, GrowOVER-QA and GrowOVER-Dialogue (**Grow**ing **O**pen-domain knowledge benchmarks for retrie**V**al-augmented gen**ER**ation). As illustrated in Figure 1, GrowOVER provides the evidence text along with the gold answers, which can be used to evaluate the retriever. In addition, we utilize the evidence text to verify the validity of previously generated datasets and maintain valid ones in the succeeding time steps. Consequently, our benchmarks continue to grow from their initial creation as new Wikipedia snapshots continue to come in. Furthermore, it introduces a dialogue task to better evaluate the generator. The open-domain dialogue task challenges models to adapt to the user's responses and potentially shift topics while still responding accurately. This demands a more sophisticated understanding and application of world knowledge, allowing for a more extensive evaluation. Table 1 presents the comparison of GrowOVER with other benchmarks.

To enable intermittently updated LLMs to cope with the rapidly evolving world, recent research has explored two approaches: retrieval (Kasai et al., 2022; Ram et al., 2023) and continual pretraining (Jang et al., 2022; Kim et al., 2023). The retrieval approach employs a retriever to supply LLMs with new information from an updated database, leveraging their in-context learning capabilities. The continual pertaining approach updates outdated knowledge within the LLMs' parameters, thereby preventing hallucinations. However, constantly updating LLMs can be costly and prone to performance degradation, while relying solely on the retriever can be vulnerable. Therefore, we propose a retrieval-interactive LLM (RiLM). In RiLM, the LLM evaluates its own answers and, if found unreliable, provides feedback to the retriever to locate more relevant documents. The LLM then uses the feedback to generate improved answers.

Finally, our contributions are as follows.

1. We introduce GrowOVER, a set of dynamic QA and dialogue benchmarks that evaluate both retrievers and generators by annotating the evidence text and introducing a challenging dialogue task.
2. We propose RiLM, a framework where the LLM evaluates its own answers and provides feedback to the retriever to correct retrievals, thereby regenerating better answers.
3. We empirically demonstrate the effectiveness of our method without requiring additional pre-training of the LLM.

2 Related Work

Temporal sensitivity. Temporal misalignment occurs when training and test datasets originate from different time periods. Past studies (Lazaridou et al., 2021; Luu et al., 2021) report poor performance in downstream tasks when making predictions beyond the training period. Thus, recent research (Dhingra et al., 2022; Liska et al., 2022; Saxena et al., 2021; Jang et al., 2022; Kim et al., 2023) efforts to evaluate how LLMs handle time-sensitive information. In particular, there have been approaches to utilize a retriever for time-sensitive knowledge. Zhang and Choi (2021) and Longpre et al. (2021) report that, even with an updated evidence corpus, language models trained on previous data struggle to respond to questions in the present. However, Kasai et al. (2022) show that LLMs can adjust their generated responses to recently retrieved documents provided by prompting. Still, when failing to retrieve appropriate documents, LLMs may produce outdated answers. To address this issue, we propose a retrieval-interactive LLM framework that allows the LLM to provide feedback to the retriever to fetch more relevant documents when the answer is less reliable.

Retrieval augmented LLM. In the initial stages, language models had limited capacity to store a vast amount of factual details. Hence, prior studies (Lewis et al., 2020; Guu et al., 2020) introduce RAG, where the generator responds based on the passages provided by the retriever. As LLMs grow larger and are pre-trained on huge text corpora, Ram et al. (2023) propose, instead of training LLMs, to combine retrieved content with a query into a prompt for LLMs to generate an answer. Additionally, Shi et al. (2023) use an ensemble scheme that provides multiple documents to the LLM, which determines the next token by summing the probability of the next token for each document. Recently, studies have focused on when or what to retrieve. For instance, Asai et al. (2023) uses special tokens to decide when to retrieve, and then generates and reflects on the passages and generated answers. Similarly, Jiang et al. (2023) generates a sentence, and if the generated tokens have low probabilities, it retrieves passages using the generated sentence for a long-form generation task.

Continual Knowledge Learning. Continual Learning (CL) focuses on training a model on multiple sequential tasks, while retaining knowledge from previously learned tasks and adapting to new ones (Chen and Liu, 2018; He et al., 2021; Chen et al., 2020; Xu et al., 2023; Hu et al., 2021; Wang et al., 2020). In the realm of knowledge-intensive tasks, there is an additional imperative for *knowledge revision*. Addressing this, Jang et al. (2021) introduce the concept of continual knowledge learning (CKL) to manage the dynamic nature of world knowledge. It involves not just retaining previous knowledge but also embracing new information and adapting to updates. These objectives align with the goal of our benchmarks.

3 The GrowOVER Dataset

GrowOVER comprises two distinct datasets: QA and Dialogue. GrowOVER-QA is designed to evaluate the ability to recall entities, while GrowOVER-Dialogue features user-expert interactions over 3-4 turns to highlight generation capabilities. Each instance from both datasets is annotated with the evidence text and the type: Unchanged, Changed, or New. Our goal is to evaluate the retention of unchanged knowledge, the updating of changed knowledge, and the acquisition of new knowledge, aligning with the objectives of CKL.

Article selection. GrowOVER is based on Wikipedia snapshots[1], which contain a vast amount of world knowledge. Although there is no limitation to applying our algorithm to generate QA and conversation for any articles, we select the articles linked to *Portal:Current Events* Wikipedia article from January 2023 to December 2023 (About 12K articles).

Overall process. We create initial QA and dialogue instances using the 2023-08-20 Wikipedia snapshot using GPT-4. For each subsequent snapshot, we label each sentence in articles as *unchanged*, *changed*, or *new* by comparing it to the previous month's snapshot. Then, we retain QA and dialogue instances when the evidence text is labeled as *unchanged*, and create new instances from *new* or *changed* sentences. This process is repeated for each month's new snapshot from September through December. Algorithms and prompt templates for sentence labeling and data generation are detailed in Appendix C. Also, the statistics of datasets are provided in Appendix E.

3.1 Initial Generation

GrowOVER-QA. Each article in snapshots in the initial month is split into paragraphs. Then,

[1]We download Wikipedia data dumps from `https://dumps.wikimedia.org/enwiki/` and use monthly snapshots from 2023-08-20 to 2023-12-20.

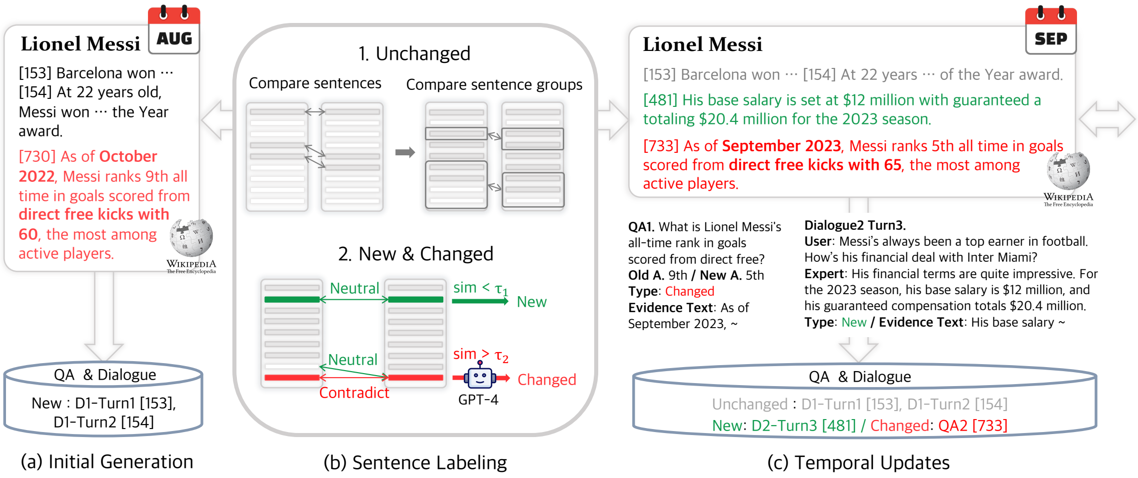

Figure 2: The overview of the dataset generation process. Please refer §3.1–3.3 together for detailed explanations.

we select up to four paragraphs, for each of which we prompt GPT-4 to generate QA. The sentences used for generating QA become the evidence text. To ensure dataset quality, we experimentally set the criteria for paragraph selection. First, we select paragraphs with less than five sentences and between 300 and 600 characters. If there are more than four satisfying paragraphs, we use K-Means clustering to group them into four clusters and randomly select one paragraph from each cluster. This creates semantically diverse QAs while avoiding too similar QAs. Afterward, we guide GPT-4 to satisfy the following: i) the question should be directly answered without the context (*e.g.* no "According to the context), ii) the answers must be short and be entities, and iii) return a bounding box indicating the sentence(s) that include the answer as the evidence text.

GrowOVER-Dialogue. The paragraph selection process is the same as QA. For each paragraph, we ask GPT-4 to create a dialogue involving user-expert interactions spanning 3-4 turns. We guide it to return the sentence used for generating each turn and then annotate it as the evidence text for each turn.

3.2 Sentence Labeling

Unchanged. Each time a new snapshot of Wikipedia becomes available, we first identify *unchanged* sentences for each pair of old and new articles. We compute sentence similarity using SimCSE (Berant et al., 2013) to localize semantically identical sentence pairs, $(s_{\text{old}}, s_{\text{new}})$. If the similarity score exceeds a threshold of 0.99, we label $(s_{\text{old}}, s_{\text{new}})$ as *unchanged*.

Additionally, we group sentences and compare them to find all unchanged sentences. Wikipedia articles generally maintain the order of sentences even after editing. Hence, we group old and new sentences into S_{old} and S_{new}, respectively, that lie between pairs of previously identified *unchanged* sentences. Then, we compute the similarity between all subsets of S_{old} and S_{new}. If any subset of S_{new} matches with a subset of S_{old} with a similarity score above the threshold, those sentences are labeled as *unchanged*. For instance, if we have matched $(s_{\text{old}_1}, s_{\text{new}_1})$ and $(s_{\text{old}_4}, s_{\text{new}_5})$ as *unchanged*, we then compare similarity between all subsets of $S_{\text{old}} = \{s_{\text{old}_2}, s_{\text{old}_3}\}$ and $S_{\text{new}} = \{s_{\text{new}_2}, s_{\text{new}_3}, s_{\text{new}_4}\}$. If the similarity score between $\{s_{\text{old}_2}\}$ and $\{s_{\text{new}_2}, s_{\text{new}_3}\}$ exceeds the threshold, we label $(s_{\text{old}_2}, \{s_{\text{new}_2}, s_{\text{new}_3}\})$ as *unchanged* as well. After identifying *unchanged* sentences in the groups, we update the groups enclosed by the new pairs of unchanged sentences. In the previous example, S_{old} becomes $\{s_{\text{old}_3}\}$ and S_{new} becomes $\{s_{\text{new}_4}\}$.

NLI. Next, we classify unlabeled sentences in S_{old} and S_{new} using a natural language inference (NLI) task with RoBERTa (Liu et al., 2019) fine-tuned on the MultiNLI dataset (Williams et al., 2018). The NLI task, given a premise and a hypothesis, classifies the hypothesis as entailment, neutral, or contradiction. In this step, we provide the model with each sentence pair $(s_{\text{old}}, s_{\text{new}})$ where $s_{\text{old}} \in S_{\text{old}}$ and $s_{\text{new}} \in S_{\text{new}}$ treating s_{old} as the premise and s_{new} the hypothesis. If s_{new} is classified as entailed by any s_{old}, we label that s_{new} as *unchanged*.

Changed. Else if s_{new} is classified as contradiction with any s_{old}, we label the $(s_{\text{old}}, s_{\text{new}})$ as *changed*. Besides, we check whether their similarity is higher than τ_2 set to 0.6 since *changed* sentences share some content, not entirely new. After that, we double-check whether s_{old} and s_{new} are contradictory with GPT-4 to ensure that $(s_{\text{old}}, s_{\text{new}})$ is *changed*.

New. Otherwise, if s_{new} is classified as neutral, we further check their similarity scores. If the similarity with all $s_{\text{old}} \in S_{\text{old}}$ is lower than τ_1 set to 0.7, we classify that s_{new} as *new* since *new* sentence should not have similar counterparts in the old document.

3.3 Temporal Updates

If an article is newly added in the new snapshot, we perform the initial generation as done in 3.1. Otherwise, based on the results of sentence labeling, we update GrowOVER in two ways: i) maintain or exclude existing QA and dialogue turn instances, and ii) generate new instances.

Maintenance. Each QA and turn instance is annotated with evidence sentences and their indices within the article. If all evidence sentences s_{old} are labeled as *unchanged*, we keep the corresponding QA or turn as Unchanged and update the evidence sentences and their indices as of the new snapshot. Otherwise, we exclude the instance since it is not guaranteed as Unchanged. For QA, we simply delete the instance from our dataset. But, for dialogue, we only exclude the turn when evaluation instead of the whole dialogue.

Generation. We generate new QA and dialogues with *new* and *changed* sentences. For QA, we find consecutive *new* sentences and split them into multiple groups if more than six sentences. For each group, we prompt GPT-4 to generate a New QA instance and annotate the evidence text as done before. For *changed* sentences, we provide GPT-4 with both the original and the revised sentences and prompt it to generate a Changed question with contradictory answers based on each sentence and annotate the revised sentences as evidence text. For dialogue generation, similarly to the initial generation process, we select informative paragraphs with *changed* or *new* sentences. We then prompt GPT-4 to generate dialogues and annotate the used sentences as the evidence text for each turn. If the evidence text is *changed* or *new*, we label the generated turn as Changed and New, respectively.

4 Approach

If the LLM has been trained on outdated data or has never been trained on new data, it may not be able to answer the questions on new information correctly. However, updating the parameters of such models should be conducted with the greatest caution to avoid potential side effects, such as catastrophic forgetting. Therefore, our approach ensures that neither the LLM nor the retriever is continuously trained with new data.

To adapt LLMs to rapidly evolving world knowledge, we propose RiLM, as shown in Figure 3. While freezing both the LLM and the retriever, we introduce the Decision Gate, which decides whether to accept the LLM's answer based on its certainty score (section 4.1). If the answer is not sufficiently confident, the retrieval-generation process is performed again, termed Adaptive Re-Retrieval (section 4.2). In this process, the LLM's previous output is fed back to the retriever to fetch documents again, enabling the LLM to generate better answers.

Similar to our approach, Asai et al. (2023) propose using reflection tokens to confirm output relevance, support, or completeness. However, their method requires training LLMs to predict reflection tokens. In contrast, RiLM only involves training the certainty classifier. Additionally, for long-form text generation, Jiang et al. (2023) generate the next sentence and then concatenate it with the question to form a retrieval query. They exclude tokens generated with low probability to avoid interrupting retrieval. Our approach also uses generated answers for re-retrieval but considers all generated tokens only to the extent that the LLM is certain about the answer.

4.1 The Decision Gate

Previous work (Ram et al., 2023) has often concatenated top-k retrieved documents $\{D_1 \cdots D_k\}$ with a query Q into a single prompt to the LLM. However, some irrelevant content may degrade performance. Thus, following Shi et al. (2023), we concatenate each document D_i with the query Q into a prompt to LLM in parallel. To select the best answer from these prompts, we add a certainty classifier to the last multi-head attention layer of the LLM.

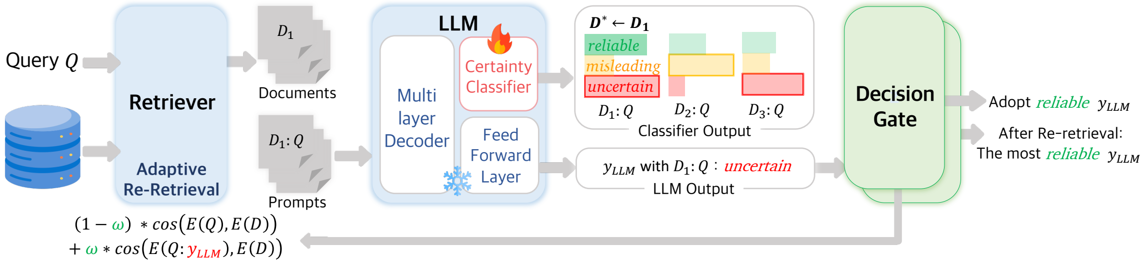

Figure 3: The RiLM framework. Given a query, we retrieve top-k documents and generate k prompts to LLM in parallel. The certainty classifier predicts either *reliable*, *misleading*, or *uncertain* for each prompt. If *reliable*, the Decision Gate adopts the answer. Otherwise, we return back to the retrieval step with LLM's output and the *reliable* probability. In Adaptive Re-retrieval, the retriever reflects this information outputs for better retrieval, based on which the LLM re-generates answers.

More specifically, we pass the query and each document through the LLM to obtain the last hidden state vector $h_{\text{LLM}}(Q, D_i)$. For each hidden state, the certainty classifier predicts either of the following three labels: i) *reliable*: the LLM confidently knows the answer, ii) *misleading*: the LLM knows the answer but could be wrong, or iii) *uncertain*: the LLM does not know the answer exactly.

To train the classifier, we assume that the LLM knows the answer if the data it has been trained on remains unchanged and the retrieval succeeds. Conversely, the LLM might incorrectly know the answer if the data it has been trained on has been updated and the retrieval fails. Lastly, the LLM does not know the answer if it has never been trained on the data and the retrieval fails. Based on this assumption, we train the classifier to predict: i) *reliable*: given UNCHANGED QA/turn with correct retrieval, ii) *misleading*: given CHANGED QA/turn with wrong retrieval, iii) *uncertain*: given NEW QA/turn with wrong retrieval. We train the classifier separately for QA and dialogue tasks. We use 512, 245, and 512 data points of UNCHANGED, CHANGED, and NEW GROWOVER-QA, respectively, and 512, 133, and 512 UNCHANGED, CHANGED, and NEW turns of GROWOVER-DIALOGUE from September.

We choose the hidden state $h_{\text{LLM}}(Q, D^*)$ with which the certainty classifier outputs the highest *reliable* probability:

$$D^* = \underset{D \,\in\, \{D_i\}}{\arg\max}\; p_{\text{CLF}}\,(\textit{reliable} \mid h_{\text{LLM}}(Q, D)). \quad (1)$$

Afterward, LLM generates the answer y_{LLM} based on the $h_{\text{LLM}}(Q, D^*)$. If the certainty classifier predicts the label as *reliable* given $h_{\text{LLM}}(Q, D^*)$, the Decision Gate adopts the answer; otherwise, we return to the retrieval step.

4.2 Adaptive Re-Retrieval

If the classifier does not predict *reliable*, we re-retrieve documents since all top-k documents are unlikely to be helpful in generating correct answers. Instead of simply retrieving the next set of top-k documents, we propose an Adaptive Re-Retrieval (ARR) method. This method feeds the LLM's answer and certainty value back to the retriever to improve relevance.

In ARR, the retriever relies on the LLM's answer to the extent that it is reliable. The *reliable* probability is computed by

$$\omega = \lambda\, p_{\text{CLF}}\,(\textit{reliable} \mid h_{\text{LLM}}(Q, D^*)). \quad (2)$$

The hyperparameter λ is set to optimize re-retrieval relevance with the training dataset for the certainty classifier. Therefore, the relevance score in the ARR is calculated as a weighted sum of two components: cosine similarity between the query and the document, and the similarity between the document and a concatenation of the query with the generated answer y_{LLM}. This is formulated as:

$$\begin{aligned} \textit{score} = {} & (1-\omega)\,\text{sim}(\mathbf{E}(Q), \mathbf{E}(D)) \\ & +\; \omega\,\text{sim}(\mathbf{E}([Q : y_{\text{LLM}}]), \mathbf{E}(D)). \end{aligned} \quad (3)$$

The retriever re-retrieves top-k documents based on Eq.(3). We adjust the reflection of the generated answer based on the LLM's predicted *reliable* probability: the less *reliable* an answer is, the less likely it is to be used. After the re-retrieval, we again choose the last hidden state vector from the document with the highest *reliable* probability among the re-retrieved documents. In the final step, the Decision Gate compares the *reliable* probabilities of the initial and newly generated answers, selecting the one with the higher probability.

5 Experiments

5.1 Experimental Setup

Baselines. We use five types of baselines: i) **Vanilla**: LLM without retrievals, ii) **Self-RAG**: an adaptive RAG baseline (Asai et al., 2023), **RaLM**: vanilla LLM with concatenated retrievals (Ram et al., 2023), iii) **RaLM-CP**: continuously pre-trained LLM (Jang et al., 2022) with concatenated retrievals, iv) **RaLM-D^***: LLM generates an answer with the classifier's selected document, and v) **RiLM**. We use top-k (k=3) documents for retrievals.

Database. Since it requires much computation to use the entire snapshot (6M articles) for the database, we randomly select 100K articles in addition to the 12K articles selected for GrowOVER generation. We first split these articles using the LangChain document loader[2], which semantically segments given documents. Then, we index documents using FAISS (Johnson et al., 2019) following Shi et al. (2023).

Retriever. Even though all retrievers can be plugged into our framework, we use Sent-BERT (Reimers and Gurevych, 2019) since it reduces time cost using Siamese encoders with strong performance. It uses cosine similarity to calculate sentence similarity. We also test Contriever (Izacard et al., 2021) and present the results in Appendix B.1. We use questions as retrieval queries for the QA task, and user queries combined with the chat history for the dialogue response task.

LLM. We use Llama2-7B (Touvron et al., 2023). We initially pre-train the model with the selected articles in the 2023-08-20 snapshot. For RaLM-CP, we continuously pre-train the model on each new snapshot available every month, using only the selected articles from these snapshots for dataset generation. For the other baselines, we freeze the initially pre-trained model. In the QA task, the prompt consists of the retrieved documents and questions, while in the dialogue task, the prompt concatenates the chat history, retrieved documents, and the user query.

Metric. We use the F1 score to evaluate QA, following Petroni et al. (2021) and use the BLEU (Papineni et al., 2002) score for the dialogue task, follwing (Chan et al., 2021).

[2]https://python.langchain.com/docs/modules/data_connection/document_loaders/.

	9	10	11	12
GrowOVER-QA				
Accuracy	79.0	75.5	75.8	74.9
Average F1 (Adopted)	53.7	52.1	53.1	52.7
Average F1 (Not-adopted)	28.2	28.5	28.8	28.6
Average F1 (ALL)	43.5	42.9	42.8	42.3
GrowOVER-Dialogue				
Accuracy	59.0	58.6	58.3	58.6
Average BLEU (Adopted)	6.03	6.11	6.17	6.13
Average BLEU (Not-adopted)	3.44	3.42	3.44	3.45
Average BLEU (ALL)	4.68	4.69	4.72	4.70

Table 2: Accuracy of the certainty classifier and results of adopted / not-adopted answers on each month.

	9	10	11	12
GrowOVER-QA				
Q	13.4	13.0	12.8	12.5
Q:y_{LLM}	14.5	14.1	13.4	13.3
ARR	**14.6**	**14.5**	**13.9**	**13.6**
GrowOVER-Dialogue				
Q	10.7	10.5	10.5	10.5
Q:y_{LLM}	11.5	11.3	11.4	11.3
ARR	**11.7**	**11.7**	**11.5**	**11.5**

Table 3: Accuracy of ARR on each month.

5.2 Experimental Results

We report not only the end-to-end performance of the QA and dialogue tasks, but also the performance of the classifier and ARR to demonstrate the effectiveness of each component. For all experiments, we exclude the training dataset of the classifier.

5.2.1 Results of the Classifier

Experimental results of the classifier are presented in Table 2. We evaluate the accuracy of all data points from all months. The accuracy is approximately 75 for GrowOVER-QA and 58 for GrowOVER-Dialogue. Since chat history is included in the prompts, it is harder for the classifier to gauge certainty based on documents and queries, so performance tends to suffer slightly. Also, to show the actual effectiveness of the classifier, we separately calculate the average F1 / BLEU scores for adopted answers and not-adopted answers in the Decision Gate. The gap between the average F1 / BLEU score of adopted and not-adopted answers is approximately 25.0 for the QA task and 2.7 for the dialogue task. This significant gap indicates the classifier can predict the certainty and reliability of LLM.

5.2.2 Results of Adaptive Re-Retrieval

We compare the accuracy of ARR and two baselines: i) search with only query (Q) and ii) always append y_{LLM} to Q (Q:y_{LLM}), each of which are using $sim(\mathbf{E}(Q), \mathbf{E}(D))$ and $sim(\mathbf{E}([Q{:}y_{\text{LLM}}]), \mathbf{E}(D))$, where $\omega = 0$ and

	9	10	11	12
New				
Vanilla	13.4	14.1	14.7	14.1
Self-RAG	23.6	22.6	23.2	22.7
RaLM	38.2	36.8	37.0	37.1
RaLM-CP†	<u>39.2</u>	<u>38.3</u>	37.6	37.7
RaLM-D^* (Ours)	37.9	38.1	<u>38.3</u>	<u>38.1</u>
RiLM (Ours)	**39.4**	**39.4**	**39.7**	**39.2**
Changed				
Vanilla	6.1	5.6	7.0	5.3
Self-RAG	18.2	19.2	17.7	20.1
RaLM	24.9	26.5	33.2	28.5
RaLM-CP†	<u>26.0</u>	**29.0**	33.6	29.6
RaLM-D^* (Ours)	25.1	<u>27.7</u>	<u>37.5</u>	<u>29.9</u>
RiLM (Ours)	**28.2**	<u>27.7</u>	**38.3**	**30.4**
Unchanged				
Vanilla	18.0	17.6	17.2	16.7
Self-RAG	26.7	25.5	25.4	26.1
RaLM	43.1	41.2	41.8	41.1
RaLM-CP†	44.0	43.3	42.7	42.0
RaLM-D^* (Ours)	<u>44.3</u>	<u>43.7</u>	<u>43.5</u>	<u>42.9</u>
RiLM (Ours)	**45.7**	**45.1**	**44.6**	**44.1**
All				
Vanilla	17.4	17.1	16.8	16.4
Self-RAG	26.3	25.1	25.1	25.7
RaLM	42.5	40.5	41.2	40.6
RaLM-CP†	43.4	42.6	42.1	41.5
RaLM-D^* (Ours)	<u>43.5</u>	<u>42.9</u>	<u>42.8</u>	<u>42.3</u>
RiLM (Ours)	**44.9**	**44.2**	**44.0**	**43.5**

†continuously pre-trained language model

Table 4: F1 scores of GrowOVER-QA on each month.

	9	10	11	12
New				
Vanilla	0.85	0.81	0.84	0.88
Self-RAG	2.37	2.29	2.36	2.21
RaLM	4.98	5.08	5.06	4.76
RaLM-CP†	5.06	5.04	5.08	4.86
RaLM-D^* (Ours)	<u>5.27</u>	<u>5.21</u>	<u>5.42</u>	<u>5.07</u>
RiLM (Ours)	**5.36**	**5.27**	**5.51**	**5.15**
Changed				
Vanilla	1.58	2.68	1.40	1.87
Self-RAG	4.31	3.74	3.28	5.00
RaLM	5.09	6.25	**6.89**	6.30
RaLM-CP†	6.11	6.98	<u>6.49</u>	6.36
RaLM-D^* (Ours)	<u>6.60</u>	<u>7.19</u>	6.01	<u>6.38</u>
RiLM (Ours)	**7.26**	**7.67**	6.05	**6.64**
Unchanged				
Vanilla	1.13	1.12	1.10	1.11
Self-RAG	2.58	2.56	2.32	2.49
RaLM	4.42	4.41	4.44	4.45
RaLM-CP†	4.40	4.43	4.43	4.45
RaLM-D^* (Ours)	<u>4.65</u>	<u>4.67</u>	<u>4.69</u>	<u>4.69</u>
RiLM (Ours)	**4.68**	**4.69**	**4.71**	**4.71**
All				
Vanilla	1.12	1.11	1.09	1.10
Self-RAG	2.58	2.55	2.32	2.48
RaLM	4.44	4.44	4.46	4.46
RaLM-CP†	4.43	4.45	4.46	4.47
RaLM-D^* (Ours)	<u>4.68</u>	<u>4.69</u>	<u>4.72</u>	<u>4.70</u>
RiLM (Ours)	**4.70**	**4.72**	**4.74**	**4.73**

†continuously pre-trained language model

Table 5: BLEU scores of GrowOVER-Dialogue on each month.

$\omega = 1$ in equation (3), respectively. The gap between ARR and the former implies the degree to which reliable y_{LLM} improves retrieval relevance, while the gap between ARR and the latter represents the degree to which ARR ignores potentially incorrect y_{LLM}. We exclude the first retrieved documents from re-retrieval to avoid using duplicate retrievals. The results are shown in table 3. In the QA task, ARR improves Q and Q:y_{LLM} by approximately 1.2 and 0.3, respectively, while in the dialogue task, the performance enhances by 1.0 and 0.2, respectively. This result indicates that the output of LLM with *reliable* probability can aid in re-retrieval. Moreover, the effect of retrieval relevance on the final answer is demonstrated in the subsequent end-to-end results.

5.2.3 Results of GrowOVER-QA

Table 4 shows the performance on GrowOVER-QA from September to December. It displays the F1 score for New, Changed, and Unchanged QAs, as well as All types. When comparing Vanilla to other baselines, it's clear that the retrieval significantly enhances performance, highlighting its crucial role in open-domain tasks. Self-RAG shows modest improvement over Vanilla but underperforms compared to other retrieval-augmented models. Also, in general, RiLM demonstrates outstanding performance over other baselines. For New, RiLM improves RaLM from 1.2 in September to 2.7 in November. It shows a higher F1 score over all months than RaLM-CP, even though RiLM is not continuously trained. For Changed, our method outperforms other baselines, except for October, when RaLM-CP exceeds our method by 1.3. Nevertheless, our RiLM shows much more robust performance across the other months, with improvements ranging from 0.8 to 4.7. Moreover, for Unchanged and All, our method surpasses the performance of other baselines. However, over several months, all baselines show performance degradation over months, indicating the need for the model update in the future.

5.2.4 Results of GrowOVER-Dialogue

Table 5 displays the consequences on GrowOVER-Dialogue from September to December. Similar to QA task, RiLM generally exhibits superior performance over other baselines. For instance, RiLM improves upon RaLM-CP by an average of 0.3 for New and 0.4 for Changed. Also, RiLM enhances performance on both the Unchanged and All datasets, indicating its overall effectiveness. However, for CHANGED in November, not only RiLM but also RaLM-D^* underperforms compared to RaLM. This performance drop only occurs on CHANGED, which may indicate the limitation in predicting *misleading* cases. From the consistent performance improvements on NEW and UNCHANGED, we can infer that LLM can identify what it knows and what it doesn't know. On the other hand, it occasionally has difficulty assuming that its answer might be wrong and verifying it.

5.2.5 Label-Based Analysis

In addition to the monthly results, we also report the average experimental results for each label across the months. Table 6 shows ARR accuracy for NEW, CHANGED, and UNCHANGED labels. For the CHANGED label in the QA task, $Q{:}y_{\text{LLM}}$ performs 1.6 points lower than Q, indicating that relying entirely on misleading y_{LLM} can significantly degrade performance. Conversely, in the dialogue task, misleading y_{LLM} for CHANGED improves performance by 1.2 points. This suggests that the detailed sentences generated in the dialogue task benefit from common knowledge between the old and new data, aiding re-retrieval. For NEW, the LLM struggles to generate detailed answers, resulting in a small gain of 0.1 points. For UNCHANGED in both tasks, y_{LLM} improves performance by 1.4 and 0.8 points, respectively, though still lower than ARR. Overall, ARR demonstrates robust and improved performance in all scenarios.

To analyze errors corrected by Decision Gate and ARR, we report the performance at each stage of the RiLM pipeline, averaged by label type, as shown in Table 7. This includes results for answers adopted by the decision gate, not adopted answers before and after ARR and after the final selection, and the final aggregated result. The classifier performs well across all types, as evidenced by the gap between adopted and not-adopted answers before ARR. The efficiency of ARR and the decision gate is highlighted by the performance improvements from before ARR to after the final selection. Although performance decreases after ARR due to the exclusion of top-k documents from re-retrieval, the final selection results for not-adopted answers improve, indicating the classifier's effectiveness in selecting more reliable answers. In the QA task, F1 scores for not-adopted answers improve by over 2.0 points across all types. In the dialogue task, BLEU scores significantly improve for NEW and CHANGED by 1.0 and 0.6 points, respectively, with little improvement in UNCHANGED.

	NEW	CHANGED	UNCHANGED
GROWOVER-QA			
Q	11.1	12.2	13.2
$Q{:}y_{\text{LLM}}$	**12.2**	10.6	14.6
ARR	**12.2**	**12.6**	**14.7**
GROWOVER-DIALOGUE			
Q	10.5	11.8	10.6
$Q{:}y_{\text{LLM}}$	10.6	13.0	11.4
ARR	**11.0**	**13.7**	**11.6**

Table 6: Accuracy of ARR for each label.

	NEW	CHANGED	UNCHANGED
GROWOVER-QA			
Adopted	50.3	39.3	52.4
Not Adopted - Before ARR	26.0	20.0	27.3
Not Adopted - After ARR	19.6	11.9	21.9
Not Adopted - Final Selection	28.7	22.0	31.0
Average	39.4	31.2	44.9
GROWOVER-DIALOGUE			
Adopted	7.02	8.37	6.08
Not Adopted - Before ARR	3.97	5.33	3.41
Not Adopted - After ARR	1.19	2.62	1.30
Not Adopted - Final Selection	4.09	5.96	3.45
Average	5.32	6.90	4.70

Table 7: F1 and BLEU scores for each label in each process of the pipeline.

6 Conclusion

To evaluate whether LLMs can adapt to the fast-evolving world knowledge, we propose GROWOVER-QA and GROWOVER-DIALOGUE. Our benchmarks annotate the evidence text and introduce dialogue tasks to evaluate retriever-augmented RaLM comprehensively. Furthermore, we suggest RiLM, an interactive retriever-generator framework by simply training a classifier for LLM to predict reliability itself. Through rigorous experiments, we show that our method can be on par with or surpass continuously pre-trained LLMs even without pre-training. However, even with retrieval, we observe that the performance degrades over time. Thus, we hope our benchmarks can be valuable resources to detect when to update the retriever or LLM in future work. Additionally, we anticipate further research into optimizing the use of retrievers to reduce the frequency of model updates.

Limitations

We highlight a few considerations for readers regarding potential limitations. We rely on various models to label each sentence in Wikipedia snapshots. Although we thoroughly designed the sentence labeling process to label sentences accurately, it can occasionally be faulty. To address this limitation, we append the MTurk study in the Appendix, showing the results are within acceptable bounds. Also, our dataset is primarily built from knowledge based on single articles, which may restrict its effectiveness for tasks that necessitate combining information from multiple sources. Further research would be beneficial to generate benchmarks that enable the evaluation of frameworks handling complex reasoning tasks with a time-sensitive nature. Moreover, the information extracted from Wikipedia may not promptly reflect real-world knowledge updates. This means that knowledge modifications in the real world might not be immediately mirrored in the dataset, resulting in some degree of outdated or inaccurate information. Finally, the characteristics of our dataset may be influenced by the features of GPT-4 that we utilized during its generation, as well as the prompts we used. To enhance the effectiveness and mitigate bias in datasets, further research should focus on generating datasets for a variety of using Large Language Models employed across diverse fields, aligned with a wide range of prompts.

Ethics Statement

We have manually reevaluated the dataset we created to ensure it is free of any potential for discrimination, human rights violations, bias, exploitation, and any other ethical concerns.

Acknowledgments

We sincerely thank Jaewoo Ahn, Soochan Lee, Yeda Song, Heeseung Yun, Junik Bae, and other anonymous reviewers for their valuable comments. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-II220156, Fundamental research on continual meta-learning for quality enhancement of casual videos and their 3D metaverse transformation), the SNU-Global Excellence Research Center establishment project, the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2023R1A2C2005573), Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(RS-2023-00274280), and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)).

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544.

Zhangming Chan, Lemao Liu, Juntao Li, Haisong Zhang, Dongyan Zhao, Shuming Shi, and Rui Yan. 2021. Enhancing the open-domain dialogue evaluation in latent space. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4889–4900.

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. *Preprint*, arXiv:2004.12651.

Zhiyuan Chen and Bing Liu. 2018. *Lifelong machine learning*, volume 1. Springer.

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.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR.

Tianxing He, Jun Liu, Kyunghyun Cho, Myle Ott, Bing Liu, James Glass, and Fuchun Peng. 2021. Analyzing the forgetting problem in the pretrain-finetuning of dialogue response models. *Preprint*, arXiv:1910.07117.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792.

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. *Preprint*, arXiv:2106.09685.

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*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2022. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models. *arXiv preprint arXiv:2204.14211*.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, KIM Gyeonghun, Stanley Jungkyu Choi, and Minjoon Seo. 2021. Towards continual knowledge learning of language models. In *International Conference on Learning Representations*.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, and Kentaro Inui. 2022. Realtime qa: What's the answer right now? *arXiv preprint arXiv:2207.13332*.

Yujin Kim, Jaehong Yoon, Seonghyeon Ye, Sung Ju Hwang, and Se-young Yun. 2023. Carpe diem: On the evaluation of world knowledge in lifelong language models. *arXiv preprint arXiv:2311.08106*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, et al. 2021. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems*, 34:29348–29363.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Adam Liska, Tomas Kocisky, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, D'Autume Cyprien De Masson, Tim Scholtes, Manzil Zaheer, Susannah Young, et al. 2022. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *International Conference on Machine Learning*, pages 13604–13622. PMLR.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. *arXiv preprint arXiv:2109.05052*.

Kelvin Luu, Daniel Khashabi, Suchin Gururangan, Karishma Mandyam, and Noah A Smith. 2021. Time waits for no one! analysis and challenges of temporal misalignment. *arXiv preprint arXiv:2111.07408*.

Katerina Margatina, Shuai Wang, Yogarshi Vyas, Neha Anna John, Yassine Benajiba, and Miguel Ballesteros. 2023. Dynamic benchmarking of masked language models on temporal concept drift with multiple views. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2873–2890.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2021. Kilt: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. 2021. Question answering over temporal knowledge graphs. *arXiv preprint arXiv:2106.01515*.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819.

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

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters. *Preprint*, arXiv:2002.01808.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *Preprint*, arXiv:2312.12148.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

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.

A Experimental Details

Dataset. For dataset generation, we use OpenAI *gpt-4-1106-preview* model and set the temperature as 0 and max_new_token as 256. Also, we use Huggingface *princeton-nlp/sup-simcse-roberta-large* for NLI task in the sentence labeling process. Additionally, when checking contradiction with GPT-4, we use *gpt-4-1106-preview* and set the temperature as 0 and max_tokens as 256.

Database. We use the Langchain document loader (RecursiveCharacterTextSplitter) to split each article in the Wikipedia snapshot into several passages. We set chunk_size as 1500 and chunk_overlap as 10 characters.

Continual Pretraining of LLM. For initial training on August, we train Llama2 for four epochs with a learning rate of 1e-06, a learning rate decay of 0.8, a cosine learning rate scheduler, an AdamW optimizer, and a batch size of 64, using FSDP (fully sharded data parallel). After that, for the c-RaLM baseline, we continuously pre-train the model with 12K articles on each month. We set all hyperparameters the same with the initial training except the epoch. To prevent catastrophic forgetting, we trained the model for only one epoch, following Jang et al. (2022).

Classifier Training. For the classifier, we trained a linear layer with dimensions (4096, 3). In both QA and dialogue response generation tasks, we set the learning rate as 0.0001, weight decay as 1e-07, and 10 epochs. We use a cosine learning rate scheduler. Also, we train the model using cross-entropy loss.

Adaptive Re-Retrieval. In both QA and dialogue tasks with SentBERT, we set the hyperparameter λ as 2.0. In the other cases, we set the λ as 1.0.

Answer Generation. We set the max_new_token of Llama2 as 10 for QA and 50 for dialogue, and load the model using bfloat16.

Self-RAG. We used short-form generation version with always_retrieve mode, since all data points of GrowOver contain evidence texts. We used same retrievals as RiLM.

B Additional Experimental Results

B.1 Contriever

We additionally perform experiments with Contriever (Izacard et al., 2021), following Shi et al. (2023) and Izacard et al. (2022). We report classifier performance, ARR results, as well as end-to-end QA results.

B.1.1 Classifier Performance

Table 8 shows the experimental results of the certainty classifier. Despite using the same certainty classifier as SentBERT, the accuracy of all cases is above 65.0, demonstrating its effectiveness. Also, across all months, F1 / BLEU scores for adopted and not-adopted answers show a significant gap.

	9	10	11	12
GrowOVER-QA				
Accuracy	77.2	74.5	74.4	72.9
Average F1 (Adopted)	28.4	27.7	27.4	27.1
Average F1 (Not-adopted)	17.0	16.7	16.5	16.2
Average F1 (ALL)	21.1	20.6	20.2	19.8
GrowOVER-Dialogue				
Accuracy	66.6	66.4	65.9	66.0
Average BLEU (Adopted)	4.44	4.44	4.46	4.46
Average BLEU (Not-adopted)	2.53	2.56	2.55	2.56
Average BLEU (ALL)	2.57	2.57	2.57	2.59

Table 8: Accuracy and F1 scores of the classifier in the Decision Gate on each month.

B.1.2 Adaptive Re-Retrieval results

As shown in Table 9, ARR significantly improves retrieval relevance. In the QA task, performance increased by more than 3 times, and in the dialogue task, it increased by more than 2 times.

	9	10	11	12
GrowOVER-QA				
Next top-k	2.6	2.6	2.5	2.6
ARR	**8.5**	**8.2**	**8.0**	**8.0**
GrowOVER-Dialogue				
Next top-k	11.6	11.6	11.6	11.6
ARR	**27.1**	**27.6**	**27.3**	**30.7**

Table 9: Adaptive Re-Retrieval relevance of Contriever compared to choosing next top-k documents on each month.

B.1.3 QA Results

The results of GrowOVER-QA are presented in Table 10. It demonstrates the robustness of our method. For New, Unchanged, and all types of QA, our RiLM shows the highest score across all months. Also, for CHANGED, RiLM surpasses the other baselines except in September. In September, the performance degrades after re-retrieval, which indicates the DG possibly struggles with selecting correct answers between two different answers. Nonetheless, across all the other months, RiLM improves RaLM-D^* about by 3.0. Also, it significantly outperforms other baselines.

	9	10	11	12
NEW				
Vanilla	13.4	14.1	14.7	14.1
RaLM	16.5	16.7	17.1	16.4
RaLM-CP[†]	17.0	17.0	17.3	16.8
RaLM-D^* (Ours)	<u>17.2</u>	<u>17.9</u>	<u>18.0</u>	<u>17.0</u>
RiLM (Ours)	**19.8**	**20.2**	**19.8**	**18.8**
CHANGED				
Vanilla	6.1	5.6	7.0	5.3
RaLM	8.1	10.7	9.0	9.2
RaLM-CP[†]	8.1	<u>11.1</u>	9.1	<u>9.3</u>
RaLM-D^* (Ours)	**11.8**	10.9	<u>10.5</u>	8.8
RiLM (Ours)	<u>11.4</u>	**13.8**	**13.5**	**12.6**
UNCHANGED				
Vanilla	18.0	17.6	17.2	16.7
RaLM	20.3	19.8	19.4	19.2
RaLM-CP[†]	20.9	20.3	19.9	19.6
RaLM-D^* (Ours)	<u>21.7</u>	<u>21.0</u>	<u>20.6</u>	<u>20.3</u>
RiLM (Ours)	**23.6**	**22.9**	**22.4**	**22.1**
ALL				
Vanilla	17.4	17.1	16.8	16.4
RaLM	19.8	19.3	19.0	18.8
RaLM-CP[†]	20.4	19.8	19.6	19.3
RaLM-D^* (Ours)	<u>21.1</u>	<u>20.6</u>	<u>20.2</u>	<u>19.8</u>
RiLM (Ours)	**23.1**	**22.5**	**22.1**	**21.7**

[†] continuously pretrained language model

Table 10: F1 score of GROWOVER-QA on each month using Contriever.

	9	10	11	12
NEW				
Vanilla	0.85	0.81	0.84	0.88
RaLM	3.28	3.11	3.22	2.95
RaLM-CP[†]	<u>3.30</u>	<u>3.13</u>	<u>3.25</u>	2.94
RaLM-D^* (Ours)	3.26	3.07	3.20	**3.10**
RiLM (Ours)	**3.61**	**3.14**	**3.66**	**3.10**
CHANGED				
Vanilla	1.58	2.68	1.40	1.87
RaLM	**5.40**	**5.39**	3.79	4.63
RaLM-CP[†]	<u>4.74</u>	<u>5.08</u>	<u>4.06</u>	4.64
RaLM-D^* (Ours)	3.94	4.66	3.92	**5.29**
RiLM (Ours)	4.28	4.71	**4.48**	**5.29**
UNCHANGED				
Vanilla	1.13	1.12	1.10	1.11
RaLM	2.50	2.53	2.52	2.54
RaLM-CP[†]	2.48	2.51	2.52	2.53
RaLM-D^* (Ours)	<u>2.54</u>	<u>2.55</u>	<u>2.54</u>	<u>2.56</u>
RiLM (Ours)	**2.66**	**2.56**	**2.64**	**2.58**
ALL				
Vanilla	1.12	1.11	1.09	1.10
RaLM	2.53	2.56	2.55	2.56
RaLM-CP[†]	2.51	2.54	2.55	2.55
RaLM-D^* (Ours)	<u>2.57</u>	<u>2.57</u>	<u>2.57</u>	<u>2.59</u>
RiLM (Ours)	**2.69**	**2.59**	**2.68**	**2.60**

[†] continuously pre-trained language model

Table 11: BLEU score of GROWOVER-DIALOGUE on each month using Contriever. The table shows the BLEU score between the generated dialogue response and the gold dialogue response.

C Algorithms

Algorithm 1 Initial Generation for GrowOVER-QA

Require: WP_{cur} := Wikipedia snapshots of the initial month
P := An empty array to store valid paragraphs
S := An empty array to store selected paragraphs
Q := An empty array to store generated QA pairs
$*$ $article$ in WP has attributes *id*, *title* and *text*

for all $article\ a_r \in WP_{init}$ **do**
  for $paragraph\ p_r \in a_r.text$ **do**
    if p_r is of adequate length **then**
      P.append(p_r)
    end if
  end for
  $S \leftarrow$ ClusterParagraphs(P)
  $Q \leftarrow Q$ + GenerateQA(S, New)
end for

function ClusterParagraphs(P)
  Extract features and obtain embeddings of P using *SimCSE*
  Compute cluster assignment using KMeans algorithm
  $S \leftarrow$ Randomly selected paragraphs from each cluster
  return S
end function

function GenerateQA(S, $type$)
  // initialize QA as an empty array
  for selected text $p \in S$ **do**
    $qa \leftarrow$ generate QA pairs with p
    $qa.type \leftarrow type$
    QA.append(qa)
  end for
  return QA
end function

Algorithm 2 Initial Generation for GrowOVER-Dialogue

Require: WP_{init} := Wikipedia snapshots at initial month
P := An empty array to store splitted paragraphs
D := An empty array to store generated Dialogue
$*$ *article* in WP has attributes *id*, *title* and *text*

for $article\ a_r \in WP_{init}$ **do**
  $P \leftarrow$ SplitArticleIntoParagraph(a_r)
  for $paragraph\ p \in P$ **do**
    if p is informative paragraph **then**
      D.append(GenerateDialogue(p, a_r.title))
    end if
  end for
end for

function SplitArticleIntoParagraph(a)
  for $paragraph\ p \in a.text$ **do**
    P.append(split p into sentence)
  end for
  return P
end function

function GenerateDialogue(p, a.title)
  $d \leftarrow$ generate dialogue with p and a.title
  for $turn\ t \in d$ **do**
    $t.type \leftarrow$ NEW
  end for
  return d
end function

Algorithm 3 Sentence Labeling

Require: `sentences_old`:= sentences in the old article, `sentences_new`:= sentences in the new article
▷ Identify *unchanged* sentence pairs
for s_{old} in `sentences_old` **do**
  for s_{new} in `sentences_new` **do**
    if `sim`(s_{old}, s_{new}) > thrs(=0.99) **then**
      (s_{old}, s_{new}) $\leftarrow$ *unchanged*
    end if
  end for
end for
▷ Identify *unchanged* sentence groups
for each group of sentences (S_{old}, S_{new}) enclosed by *unchanged* pairs **do**
  for sub_{old} in $\mathcal{P}(S_{old})$ **do**
    for sub_{new} in $\mathcal{P}(S_{old})$ **do**
      if `sim` (`concat`(sub_{old}), `concat`(sub_{new})) > thrs(=0.99) **then**
        (sub_{old}, sub_{new}) $\leftarrow$ *unchanged*
      end if
    end for
  end for
end for
▷ NLI for *changed* & *new*
for each group of sentences (S_{old}, S_{new}) enclosed by *unchanged* pairs **do**
  for s_{new} in S_{new} **do**
    `preds` $\leftarrow$ an empty list
    for s_{old} in S_{old} **do**
      `preds`.append(`NLI.classify`(s_{old}, s_{new}))
    end for
    if "entailment" in `preds` **then**
      $s_{old} \leftarrow$ the entailed old sentence
      (s_{old}, s_{new}) $\leftarrow$ *unchanged*
    else if "contradiction" in `preds` **then**
      $s_{old} \leftarrow$ the contradicted old sentence
      if `sim` (s_{old}, s_{new}) > τ_1 (=0.6) **then**
        if `GPT-4.contradict`(s_{old}, s_{new}) **then**
          (s_{old}, s_{new}) $\leftarrow$ *changed*
        end if
      end if
    else ▷ all elements in `preds` are "neutral"
      `sim_res` $\leftarrow$ [`sim`(s_{old}, s_{new}) for s_{old} in S_{old})]
      `max_similarity` $\leftarrow$ `max`(`sim_res`)
      if `max_similarity` < τ_2 (=0.7) **then**
        $s_{new} \leftarrow$ *new*
      end if
    end if
  end for
end for

Algorithm 4 Temporal Updates for GrowOVER-QA

```
Require: W_t := Wikipedia snapshots of month t
QA_{t-1} := QAs of month t-1
QA_t := An empty set for QAs of month t

for article a_t ∈ W_t do
    for a_t's QA qa ∈ QA_{t-1} do
        qa ← UpdateQA(qa)
        if qa.type is Unchanged then
            QA_t.append(qa)
        end if
    end for
    G_new ← groups of new sentences in a_t
    for group g ∈ G_new do
        QA_t.append(GenerateQA(g, New))
    end for
    for changed sentence s_t ∈ a_t do
        s_{t-1} ← contradictory sentence in a_{t-1}
        QA_t.append(GenerateQA(s_{t-1}, s_t, Changed))
    end for
end for

function UpdateQA(qa)
    if all sentences ∈ qa.evid_text are unchanged then
        qa.type ← Unchanged
        qa.index ← indices of qa.evid_text in W_t
    else
        qa.type ← Deleted
    end if
    return qa
end function
```

Algorithm 5 Temporal Updates for GrowOVER-Dialogue

```
Require: W_t := Wikipedia snapshots on month t
D_{t-1} := Dialogues of month t-1
D_t := An empty set for dialogues of month t

for article a_t ∈ W_t do
    for a_t's dialogue d ∈ D_{t-1} do
        d ← UpdateDialog(d, L_{t-1})
        D_t.append(d)
    end for
    P ← SplitArticleIntoParagraph(a_t)
    for paragraph p ∈ P do
        if p is not informative paragraph then
            continue
        else if changed or new sentence in p then
            D_t.append(GenerateDial(p, a_t.title))
        end if
    end for
end for

function UpdateDialog(d)
    for turn t ∈ d do
        if s is unchanged sentence then
            t.type ← Unchanged
            t.index ← the index of t.evid_text in WP_cur
        else
            t.type ← Deleted
        end if
    end for
    return d
end function
```

D Mturk Study

We thoroughly designed the sentence labeling process to accurately label sentences (e.g., setting a high threshold for selecting semantically identical sentences). To further validate using sentence similarity scores and natural language inference, we employed Amazon Mechanical Turk (AMT) workers to assess the sentence labels during the rebuttal process.

We randomly sampled 30 new and 30 changed sentences and asked AMT workers to classify whether each sentence in the new article was supported, not supported, or uncertain given the old article. For new sentences, the labeling is incorrect when classified as "supported," while for changed sentences, the labeling is correct when classified as "not supported." Each sentence was evaluated by three workers, and the majority vote was used. The human quality check results are shown in the table 12.

Although new sentences have lower accuracy since they are not verified with GPT-4, the results are still within acceptable bounds. Evaluating the correct label by humans requires a review of the entire article, which is extremely time-consuming. This becomes critical, especially for our dataset, which requires regular updates to reflect the ever-changing knowledge of the real world. Therefore, as evidenced by the human quality check results, our fully automated sentence labeling alone can efficiently provide reasonably accurate labels with no human effort at all.

	New	Changed
GrowOVER-QA		
Accuracy	86.7	96.7

Table 12: Sentence labeling validation.

E Data Analysis

	Unchanged	New	Changed	Deleted	Total
08	-	32,807	-	-	32,807
09	32,422 (512)	4,936 (512)	290 (245)	385	37,648 (1,269)
10	36,863	5,193	307	785	42,363
11	41,257	5,363	309	1,106	46,929
12	43,422	5,082	313	1,211	48,817

Table 13: QA

	Unchanged	New	Changed	Deleted	Total
08	-	108,128	-	-	108,128
09	109,752 (512)	4,478 (512)	156 (133)	987	114,386 (1,157)
10	115,022	4,797	147	1,883	119,966
11	120,551	4,870	158	2,161	125,579
12	125,940	5,218	142	2,427	131,300

Table 14: Dialogue

F Dataset Examples

Table 15 shows the examples of QA. Table 16 and 17 show the examples of Dialogue.

G Case Study

Table 18 and Table 19 present a case study from GrowOVER-QA. **Predictions** denote the answers generated by each retriever-generator framework. Table 20 presents a case study from GrowOVER-Dialogue.

H Prompt examples

Table 21 shows the prompt used for the initial generation of the QA pair. Table 22 and 23 show the prompts used to generate New QA pairs, without and with Source Content respectively. Table 24 shows the prompt used to generate Changed QA pairs. Table 25 shows the prompt used for Dialogue generation.

Article : Politics of Cambodia	
Type : CHANGED	
Question: Who is the current prime minister of Cambodia from the Cambodian People's Party (CPP)?	
Answer: Hun Manet	
Previous Answer: Hun Sen	
Evidence Text: The current prime minister is Cambodian People's Party (CPP) member Hun Manet.	
Evidence Index: 80	
Article : The Eras Tour	
Type : NEW	
Question: Who is directing the concert film *Taylor Swift: The Eras Tour*?	
Answer: Sam Wrench	
Evidence Text: On August 31, 2023, Swift announced the concert film *Taylor Swift: The Eras Tour*, directed by Sam Wrench. Recorded at SoFi Stadium in Los Angeles, the film is scheduled for release to theaters in North America on October 13.	
Evidence Index: 263	
Article : Airline	
Type : UNCHANGED	
Question: Which airport would most Manhattan travelers prefer for its proximity?	
Answer: LaGuardia Airport	
Evidence Text: For example, LaGuardia Airport is the preferred airport for most of Manhattan due to its proximity, while long-distance routes must use John F. Kennedy International Airport's longer runways.	
Evidence Index: 297	

Table 15: An Example of QA

Article : 2030 Winter Olympics
Paragraph : [20]A decision on the 2030 Winter Olympics host city has been delayed until 23 July 2024, to allow the IOC more time to carefully plan the future of the Winter Olympics. [21]Due to impacts from climate change, the IOC is considering a number of changes to future games, such as rotating host cities, limiting the games to existing or temporary venues, and establishing minimum average temperature requirements. [22]It also considered awarding the 2030 and 2034 winter games simultaneously at the next IOC session, but IOC President Thomas Bach has ruled out that possibility because "it would not be the right thing to do". [23]However, at the 141st IOC Session in Mumbai, where these games were originally set to be awarded, the IOC approved to award both the 2030 and 2034 Winter Olympics at the 142nd IOC Session in Paris on the eve of the 2024 Summer Olympics.
Dialogue :
Type: UNCHANGED **User**: I was looking forward to finding out which city will host the 2030 Winter Olympics, but I heard there's been a delay. What's going on with that? **Expert**: Yes, that's correct. The decision has been postponed until 23 July 2024. The International Olympic Committee (IOC) wants to take more time to plan the future of the Winter Games with careful consideration. **Evidence Index**: 20
Type: UNCHANGED **User**: I guess the delay must be frustrating for the candidate cities. I've heard that the changing climate is causing some issues? **Expert**: Indeed, the changing climate is a significant concern. The IOC is exploring various adjustments for future games, including the possibility of having rotating host cities and using existing or temporary venues to mitigate the impact. **Evidence Index**: 21
Type: UNCHANGED **User**: I also caught a rumor that they might announce the hosts for both 2030 and 2034 at the same time. Is that true? **Expert**: That was a consideration at one point. However, IOC President Thomas Bach has stated that awarding both the 2030 and 2034 Winter Olympics simultaneously wouldn't be appropriate. **Evidence Index**: 22
Type: NEW **User**: So, when can we expect the announcement for the 2030 host city? **Expert**: The announcement is set to be made at the 142nd IOC Session in Paris, which will happen right before the 2024 Summer Olympics. They will also award the 2034 Winter Olympics at that session. **Evidence Index**: 23

Table 16: An Example of Dialogue

Article : Tornadoes of 2023	
Paragraph : [357]More rounds of severe storms occurred on June 17, including a powerful MCS that pushed through Northern and Central Oklahoma. [358]Both the Oklahoma City and Tulsa metropolitan areas had widespread wind damage with the latter metropolitan area seeing wind gusts of up to and multiple weak tornadoes. [359]On June 18, more tornadoes were confirmed in Florida and Mississippi, including an EF3 tornado which caused significant damage in Louin, Mississippi, where at least one fatality and 20 injuries occurred. [360]Scattered tornadoes occurred across the Southeast on June 19 as well, including an EF2 tornado that injured six people in Moss Point, Mississippi. [361]Overall, 87 tornadoes were confirmed during this outbreak sequence.	
Dialogue :	
Type: Unchanged **User**: I heard there were some intense storms in Oklahoma last June. What happened exactly? **Expert**: Indeed, on June 17, Oklahoma experienced severe storms, including a particularly strong Mesoscale Convective System that affected the northern and central parts of the state. **Evidence Index**: 357	
Type: Unchanged **User**: That sounds serious. Did the storms cause a lot of damage in the area? **Expert**: Yes, the storms led to extensive wind damage, especially in the Oklahoma City and Tulsa areas. Tulsa even experienced wind gusts of up to 80 mph and several weak tornadoes. **Evidence Index**: 358	
Type: Unchanged **User**: I also caught wind of some tornado activity in other states around that time. **Expert**: That's correct. The following day, on June 18, Florida and Mississippi reported tornadoes. Mississippi was hit by an EF3 tornado in Louin, which resulted in one death and 20 injuries. **Evidence Index**: 359	
Type: Unchanged **User**: Was that the end of it, or did the tornadoes continue? **Expert**: The tornadoes continued. On June 19, the Southeast saw more, including an EF2 tornado in Moss Point, Mississippi, which injured six people. **Evidence Index**: 360	
Type: Changed **User**: It must have been a large outbreak to affect so many areas. **Expert**: It was indeed a significant outbreak. In total, there were 87 confirmed tornadoes during that sequence of storms. **Evidence Index**: 361	

Table 17: Another Example of Dialogue

Article : Football player

Type : Unchanged

Question: What was the average annual salary for goalkeepers in Major League Soccer during the 2013 season?

Answer: $85,296

Evidence Text: For example, the average annual salary for footballers in Major League Soccer for the 2013 season was $148,693, with significant variations depending on the player position (goalkeepers for example earned $85,296, whereas forwards earned $251,805.

Predictions :

Vanilla: $110,000 **RaLM**: $148,693 **RaLM-CP**: $148,693 **RaLM-$D^*$ (Ours)**: $85,296 **RiLM (Ours)**: $85,296

Retrieved Documents :

[Top-3]

(1) ***Hit***: However, only a fraction of men's professional football players is paid at this level. ... **For example, the average annual salary for footballers in Major League Soccer for the 2013 season was $148,693, with significant variations depending on the player position (goalkeepers for example earned $85,296, whereas forwards earned $251,805.** Popularity and average salaries in women's leagues are far lower. For example, players in ...

(2) *Miss*: MLS has a set of pool goalkeepers who are signed to a contract with the league and are loaned to teams during emergencies in which they are missing a goalkeeper due to injuries or suspensions. ... These initiatives have brought about an increase in on-field competition.

(3) *Miss*: According to "France Football", Messi was the world's highest-paid footballer for five years out of six between 2009 and 2014; ... In 2020, Messi became the second footballer, as well as the second athlete in a team sport, after Cristiano Ronaldo, to surpass $1 billion in earnings during their careers.

[RaLM-D^*]

Hit: However, only a fraction of men's professional football players is paid at this level. ... **For example, the average annual salary for footballers in Major League Soccer for the 2013 season was $148,693, with significant variations depending on the player position (goalkeepers for example earned $85,296, whereas forwards earned $251,805.** Popularity and average salaries in women's leagues are far lower. For example, players in ...

[RiLM]

Hit: However, only a fraction of men's professional football players is paid at this level. ... **For example, the average annual salary for footballers in Major League Soccer for the 2013 season was $148,693, with significant variations depending on the player position (goalkeepers for example earned $85,296, whereas forwards earned $251,805.** Popularity and average salaries in women's leagues are far lower. For example, players in ...

Article : Benjamin Netanyahu

Type : New

Question: What city was Benjamin Netanyahu born in?

Answer: Tel Aviv

Evidence Text: Netanyahu was born in Tel Aviv, to Benzion Netanyahu (original name Mileikowsky) and Tzila (Cela;

Predictions :

Vanilla: Jerusalem, Israel **RaLM**: Tel Aviv **RaLM-CP**: Tel Aviv **RaLM-D^* (Ours)**: Tel Aviv **RiLM (Ours)**: Tel Aviv

Retrieved Documents :

[Top-3]

(1) *Miss*: Netanyahu was the second of three children. He was initially raised and educated in Jerusalem, where he attended ... the liberal sensibilities of the Reform synagogue, Temple Judea of Philadelphia, that the family attended.

(2) ***Hit***: **Netanyahu was born in Tel Aviv, to Benzion Netanyahu (original name Mileikowsky) and Tzila (Cela;** née Segal). His mother was born in 1912 in Petah Tikva, then in Ottoman Palestine, now Israel. Though all his grandparents were born in ...

(3) *Miss*: Netanyahu made his closeness to Donald Trump, a personal friend since the 1980s, central to his political appeal in Israel from 2016. ... He claims descent from the Vilna Gaon.

[RaLM-D^*]

Hit: **Netanyahu was born in Tel Aviv, to Benzion Netanyahu (original name Mileikowsky) and Tzila (Cela;** née Segal). His mother was born in 1912 in Petah Tikva, then in Ottoman Palestine, now Israel. Though all his grandparents were born in ...

[RiLM]

Hit: **Netanyahu was born in Tel Aviv, to Benzion Netanyahu (original name Mileikowsky) and Tzila (Cela;** née Segal). His mother was born in 1912 in Petah Tikva, then in Ottoman Palestine, now Israel. Though all his grandparents were born in ...

Table 18: Case Study for QA

Article : Kyrylo Budanov

Type : CHANGED

Question: What is Kyrylo Budanov's military rank?

Answer: Lieutenant general

Evidence Text: He holds the rank of lieutenant general.

Predictions :

Vanilla: Kyrylo Budanov is a Major General in the Ukrainian Armed Forces. **RaLM**: Lieutenant General **RaLM-CP**: Lieutenant General **RaLM-D^* (Ours)**: Lieutenant General **RiLM (Ours)**: Lieutenant General

Retrieved Documents :

[Top-3]

(1) ***Hit***: Kyrylo Oleksiyovych Budanov (; born 4 January 1986) is a Ukrainian military leader who is the chief of the ... Budanov previously served as the Deputy Director of one of the Departments of the Foreign Intelligence Service of Ukraine. **He holds the rank of lieutenant general.** ... as head of the Main Intelligence Directorate of the Ministry of Defense.

(2) *Miss*: The Austrian military has a wide variety of equipment. Recently, Austria has spent considerable amounts of money modernizing its military arsenal. ...

(3) *Miss*: Soon after the start of the German invasion of the Soviet Union, he was soon re-drafted into the Red Army on 4 July 1941 and initially deployed to the front as part of the 50th Cavalry Regiment. ...

[RaLM-D^*]

Hit: Kyrylo Oleksiyovych Budanov (; born 4 January 1986) is a Ukrainian military leader who is the chief of the ... Budanov previously served as the Deputy Director of one of the Departments of the Foreign Intelligence Service of Ukraine. **He holds the rank of lieutenant general.** ... as head of the Main Intelligence Directorate of the Ministry of Defense.

[RiLM]

Hit: Kyrylo Oleksiyovych Budanov (; born 4 January 1986) is a Ukrainian military leader who is the chief of the ... Budanov previously served as the Deputy Director of one of the Departments of the Foreign Intelligence Service of Ukraine. **He holds the rank of lieutenant general.** ... as head of the Main Intelligence Directorate of the Ministry of Defense.

Article : Darwin, Northern Territory

Type : UNCHANGED

Question: What is the name of the passenger train service that connects Darwin to Adelaide?

Answer: The Ghan

Evidence Text: The first service ran in 2004. "The Ghan" passenger train service from Adelaide via Alice Springs and Katherine runs once per week in each direction, with some exceptions.

Predictions :

Vanilla: The Ghan **RaLM**: The Ghan **RaLM-CP**: The Ghan **RaLM-D^* (Ours)**: The Ghan **RiLM (Ours)**: The Ghan

Retrieved Documents :

[Top-3]

(1) ***Hit***: The Territory's public transport services are managed by the Department of Lands and Planning, ... **The first service ran in 2004. "The Ghan" passenger train service from Adelaide via Alice Springs and Katherine runs once per week in each direction, with some exceptions.** Historically, ...

(2) *Miss*: Darwin and its suburbs, Palmerston and Katherine, are powered by the Channel Island Power Station, the Northern Territory's largest power plant, and the ...

(3) *Miss*: The platform layout at Flinders Street is unusual among Australian terminal stations for being almost entirely composed of through tracksa product ...

[RaLM-D^*]

Hit: The Territory's public transport services are managed by the Department of Lands and Planning, ... **The first service ran in 2004. "The Ghan" passenger train service from Adelaide via Alice Springs and Katherine runs once per week in each direction, with some exceptions.** Historically, ...

[RiLM]

Hit: The Territory's public transport services are managed by the Department of Lands and Planning, ... **The first service ran in 2004. "The Ghan" passenger train service from Adelaide via Alice Springs and Katherine runs once per week in each direction, with some exceptions.** Historically, ...

Table 19: Another Case Study for QA

Article : Marshall Islands

Turn 1

Type : Changed

User: I've heard that the Marshall Islands were inhabited for a very long time. What kind of influence did the early settlers have on these islands?

Expert: Indeed, the islands have been inhabited since the 2nd millennium BC. The early Austronesian settlers brought with them Southeast Asian crops like coconuts and breadfruit, as well as domesticated chickens, which were crucial for making the islands habitable.

Evidence Text: Austronesian settlers reached the Marshall Islands as early as the 2nd millennium BC and introduced Southeast Asian crops, including coconuts, giant swamp taro, and breadfruit, as well as domesticated chickens which made the islands permanently habitable.

Predictions :

Vanilla: The Marshall Islands were inhabited for a very long time. The early settlers had a big influence on the islands. They built houses, farmed, and fished. They also made tools and weapons. The early settlers also had a big influence on the culture of the Marshall Islands. They taught the people how to make clothes, how to cook food, and how to make music. The early settlers also taught the people how to make weapons and how to fight. The early settlers

RaLM: The early settlers of the Marshall Islands introduced Southeast Asian crops, including coconuts, giant swamp taro, and breadfruit, as well as domesticated chickens which made the islands permanently habitable. They developed a complex technique for interisland piloting by reading disruptions in ocean swells caused by low-lying coral atolls.

RaLM-CP: The early settlers of the Marshall Islands introduced Southeast Asian crops, including coconuts, giant swamp taro, and breadfruit, as well as domesticated chickens which made the islands permanently habitable. They developed a complex technique for interisland piloting by reading disruptions in ocean swells caused by low-lying coral atolls.

RaLM-D^* (Ours): The early settlers introduced Southeast Asian crops, including coconuts, giant swamp taro, and breadfruit, as well as domesticated chickens which made the islands permanently habitable.

RiLM (Ours): The early settlers introduced Southeast Asian crops, including coconuts, giant swamp taro, and breadfruit, as well as domesticated chickens which made the islands permanently habitable.

Retrieved Documents :

[Top-3]

(1) *Miss*: The majority of the citizens of the Republic of Marshall Islands are of Marshallese descent, ... around the 1st century CE at the village of Laura on Majuro and on Kwajalein Atoll.

(2) ***Hit***: **Austronesian settlers reached the Marshall Islands as early as the 2nd millennium BC and introduced Southeast Asian crops, including coconuts, giant swamp taro, and breadfruit, as well as domesticated chickens which made the islands permanently habitable.** They developed a complex technique for ...

(3) *Miss*: The British sea captains John Marshall and Thomas Gilbert visited the islands in 1788. Their vessels had been part of the First Fleet taking convicts from England to Botany Bay ...

[RaLM-D^*]

Hit: **Austronesian settlers reached the Marshall Islands as early as the 2nd millennium BC and introduced Southeast Asian crops, including coconuts, giant swamp taro, and breadfruit, as well as domesticated chickens which made the islands permanently habitable.** They developed a complex technique for ...

[RiLM]

Hit: **Austronesian settlers reached the Marshall Islands as early as the 2nd millennium BC and introduced Southeast Asian crops, including coconuts, giant swamp taro, and breadfruit, as well as domesticated chickens which made the islands permanently habitable.** They developed a complex technique for ...

Turn 2

Type : New

User: That's fascinating. I wonder how they navigated between the islands back then.

Expert: They actually developed a sophisticated method for navigating the ocean. They could read the disruptions in ocean swells, which were caused by the coral atolls, to pilot between the islands.

Evidence Text: They developed a complex technique for interisland piloting by reading disruptions in ocean swells caused by low-lying coral atolls.

Predictions :

Vanilla: The early settlers developed a complex technique for interisland piloting by reading disruptions in ocean swells caused by low-lying coral atolls.

RaLM: The early settlers developed a complex technique for interisland piloting by reading disruptions in ocean swells caused by low-lying coral atolls.

RaLM-CP: The early settlers developed a complex technique for interisland piloting by reading disruptions in ocean swells caused by low-lying coral atolls.

RaLM-D^* (Ours): The early settlers developed a complex technique for interisland piloting by reading disruptions in ocean swells caused by low-lying coral atolls.

RiLM (Ours): The early settlers developed a complex technique for interisland piloting by reading disruptions in ocean swells caused by low-lying coral atolls.

Retrieved Documents :

[Top-3]

(1) ***Hit***: ... which made the islands permanently habitable. **They developed a complex technique for interisland piloting by reading disruptions in ocean swells caused by low-lying coral atolls.** In 1526, Spanish explorer Alonso de Salazar ...

(2) *Miss*: The majority of the citizens of the Republic of Marshall Islands are of Marshallese descent, ... around the 1st century CE at the village of Laura on Majuro and on Kwajalein Atoll.

(3) *Miss*: The Austronesian settlers introduced Southeast Asian crops, including coconuts, giant swamp taro, and breadfruit, as well as domesticated chickens throughout the Marshall Islands. ...

[RaLM-D^*]

Hit: ... which made the islands permanently habitable. **They developed a complex technique for interisland piloting by reading disruptions in ocean swells caused by low-lying coral atolls.** In 1526, Spanish explorer Alonso de Salazar ...

[RiLM]

Hit: ... which made the islands permanently habitable. **They developed a complex technique for interisland piloting by reading disruptions in ocean swells caused by low-lying coral atolls.** In 1526, Spanish explorer Alonso de Salazar ...

Turn 3

Type : UNCHANGED

User: I guess that means they were pretty isolated until Europeans arrived.

Expert: That's correct. The first European who might have sighted the islands was Spanish explorer Alonso de Salazar in 1526, and the first confirmed European landing was by Álvaro de Saavedra Cerón two years later on an uninhabited island.

Evidence Text: In 1526, Spanish explorer Alonso de Salazar may have been the first European to sight the islands, and Álvaro de Saavedra Cerón made landfall on an uninhabited island two years later.

Predictions :

Vanilla: That's correct. The Marshall Islands were not discovered by Europeans until 1529, when Spanish explorer Alvaro de Mendana de Neira arrived. He named the islands after his patron, the Spanish admiral Álvaro de Mendaña de Neyra.

RaLM: That's right. The islands were largely untouched by European explorers until the late 16th century.

RaLM-CP: That's right. The islands were largely untouched by European explorers until the late 16th century.

RaLM-D^* (Ours): That's correct. The islands were largely untouched by European explorers until the 18th century.

RiLM (Ours): That's correct. The islands were largely untouched by European explorers until the 18th century.

Retrieved Documents :

[Top-3]

(1) ***Hit***: ... by low-lying coral atolls. **In 1526, Spanish explorer Alonso de Salazar may have been the first European to sight the islands, and Álvaro de Saavedra Cerón made landfall on an uninhabited island two years later.** Several other Spanish ...

(2) *Miss*: The majority of the citizens of the Republic of Marshall Islands are of Marshallese descent, ... around the 1st century CE at the village of Laura on Majuro and on Kwajalein Atoll.

(3) *Miss*: The British sea captains John Marshall and Thomas Gilbert visited the islands in 1788. Their vessels had been part of the First Fleet taking convicts from England to Botany Bay in New South Wales, ...

[RaLM-D^*]

Hit: ... by low-lying coral atolls. **In 1526, Spanish explorer Alonso de Salazar may have been the first European to sight the islands, and Álvaro de Saavedra Cerón made landfall on an uninhabited island two years later.** Several other Spanish ...

[RiLM]

Hit: ... by low-lying coral atolls. **In 1526, Spanish explorer Alonso de Salazar may have been the first European to sight the islands, and Álvaro de Saavedra Cerón made landfall on an uninhabited island two years later.** Several other Spanish ...

Turn 4
Type : Unchanged
User: And how did the Marshall Islands get their name?
Expert: The name comes from the British captains John Marshall and Thomas Gilbert who landed on Mili Atoll in 1788. It was the first recorded European presence in over 200 years, and subsequent maps began to refer to the islands by Captain Marshall's name.
Evidence Text: Later maps and charts named the islands after Captain Marshall.

Predictions :
Vanilla: The islands were named after British explorer John Marshall, who was the first European to map them in 1788.
RaLM: The islands were named after Captain John Marshall, who was the first European to sight them.
RaLM-CP: The islands were named after Captain John Marshall, who was the first European to sight them.
RaLM-D^* (Ours): The islands were named after Captain John Marshall, who was the first European to map the islands in 1788.
RiLM (Ours): The islands were named after Captain John Marshall, who was the first European to map the islands in 1788.

Retrieved Documents :
[Top-3]
(1) ***Hit***: ... in the archipelago in over 200 years. **Later maps and charts named the islands after Captain Marshall.** ...
(2) *Miss*: The majority of the citizens of the Republic of Marshall Islands are of Marshallese descent, ... around the 1st century CE at the village of Laura on Majuro and on Kwajalein Atoll.
(3) *Miss*: The British sea captains John Marshall and Thomas Gilbert visited the islands in 1788. Their vessels had been part of the First Fleet taking convicts from England to Botany Bay in New South Wales, ...

[RaLM-D^*]
Hit: ... in the archipelago in over 200 years. **Later maps and charts named the islands after Captain Marshall.** ...

[RiLM]
Hit: ... in the archipelago in over 200 years. **Later maps and charts named the islands after Captain Marshall.** ...

Table 20: Case Study for Dialogue

Generate a Q&A pair based on a given context, where the context is understood but NOT DIRECTLY VISIBLE to the person answering the question. Assume the person answering the question has common sense and is aware of the details and key points in the paragraph, but the paragraph itself is not quoted or referenced directly.

Paragraph (a list of sentences): {*paragraph*}

Use the following instructions for generating a Q&A pair:
1) Provide a question, an answer, and a bounding box.
2) DON'T use phrases such as 'according to the paragraph' in your question.
3) An answer should be an entity or entities. Provide a SHORT ANSWER.
4) The bounding box for a paragraph is defined as (starting sentence index, ending sentence index): the bounding box should be sufficiently large to encompass all the information necessary for a reader to FULLY infer the answer to the question.
5) The sentence index starts from 0.
6) Generate a SINGLE Q&A pair.

Be sure to follow the following format and write your answer within curly brackets.
The format is as follows:
{Question}{Answer}{starting sentence index}{ending sentence index}

Table 21: Sample prompt for initial generation of a QA pair

Generate a Q&A pair based on a given context, where the context is understood but NOT DIRECTLY VISIBLE to the person answering the question. Assume the person answering the question has common sense and is aware of the details and key points in the sentence(s), but the sentence(s) itself is not quoted or referenced directly.

Sentence(s): {*sentences*}

Use the following instructions for generating a Q&A pair:
1) Provide a question, and an answer.
2) DON'T use phrases such as 'according to the sentence(s)' in your question.
3) An answer should be an entity or entities. Provide a SHORT ANSWER.
4) Generate a SINGLE Q&A pair.

Be sure to follow the following format and write your answer within curly brackets.
The format is as follows:
{Question}{Answer}

Table 22: Sample prompt for generation of NEW QA pair (1)

Generate a Q&A pair based on New Sentence(s), where the context in understood but NOT DIRECTLY VISIBLE to the person answering the question. You can reference the Source Content for broader context, but the Q&A pair should relate directly to the information in New Sentence(s).

New Sentence(s): {*sentences*}

Source Content : {*source content*}

Use the following instructions for generating a Q&A pair:
1) Provide a question, and an answer.
2) DON'T use phrases such as 'according to the sentence(s)' in your question.
3) An answer should be an entity or entities. Provide a SHORT ANSWER.
4) Generate a SINGLE Q&A pair.

Be sure to follow the following format and write your answer within curly brackets.
The format is as follows:
{Question}{Answer}

Table 23: Sample prompt for generation of NEW QA pair (2)

Identify the contradiction between two following sentences and generate a Q&A pair that reflects this contradiction. The question should be answerable based on each sentence(s), but the two answers should CONTRADICT EACH OTHER. You can reference the Source Content for broader context, but the Q&A pair should relate directly to the information in Old/New Sentence(s).

Old Sentence(s) : {*old sentence*}

New Sentence(s) : {*new sentence*}

Source Content : {*source content*}

Use the following instructions for generating a Q&A pair:

1) The question should be answerable based on each sentence.

2) DON'T use phrases such as 'according to the sentence(s)' in your question.

3) An answer should be an entity or entities. Provide a SHORT ANSWER.

4) Create a SINGLE Q&A pair, providing two CONTRADICTORY answers: one based on the old sentence, and another based on the new sentence.

Be sure to follow the following format and write your answer within curly brackets.

The format is as follows:

{Question}{Answer based on Old Sentence}{Answer based on New Sentence}

Table 24: Sample prompt for generation of CHANGED QA pair

Create an Information Dialogue Dataset about *{topic}* between two conversation partners (User, Expert).

A paragraph about *{topic}* will be provided as factual information. The expert's words must be generated to provide an answer based on this information.

Using the following instruction for generating a dialogue:

1) The user starts the dialogue first

2) Create a multi-turn dialogue of 3-4 turns, each consisting of a not too long conversation.

3) Create it to include each element of conversation, discussion, and QA. In other words, users should not always ask questions using interrogative sentences.

4) DON'T use phrases such as according to the paragraph in guide's utterance.

5) DON'T simply parrot this paragraph or referenced directly. There is no need to include everything given in the paragraph in the dialogue.

6) Do not use what you already know about *{topic}*, and the Expert will answer only with the content of the provided paragraph.

7) I will provide you with sentences and a unique number for each sentence. You must indicate the Sentence number you've referenced for each turn.

Below is an example of output format and dialogues:

{{Reference Sentence}}2{{User}}I really love Granny Smith apples, they're my favorite type of apple{{Expert}}I love granny smith apples. they have hard, light green skin and a crisp flesh.

{{Reference Sentence}}1{{User}}Yes, I really enjoy them. I also like Honeycrisp apples but they're so expensive!{{Expert}}they've been grown for thousands of years in asia and europe, and were brought to north america by european colonists

{{Reference Sentence}}3{{User}}Oh really? They've been around way longer than I thought!{{Expert}}they're also consumed raw, it's one of the most popular cooking apples.

Sentences:

{sentences}

Please generate dialogue:

Table 25: Sample prompt for Dialogue generation

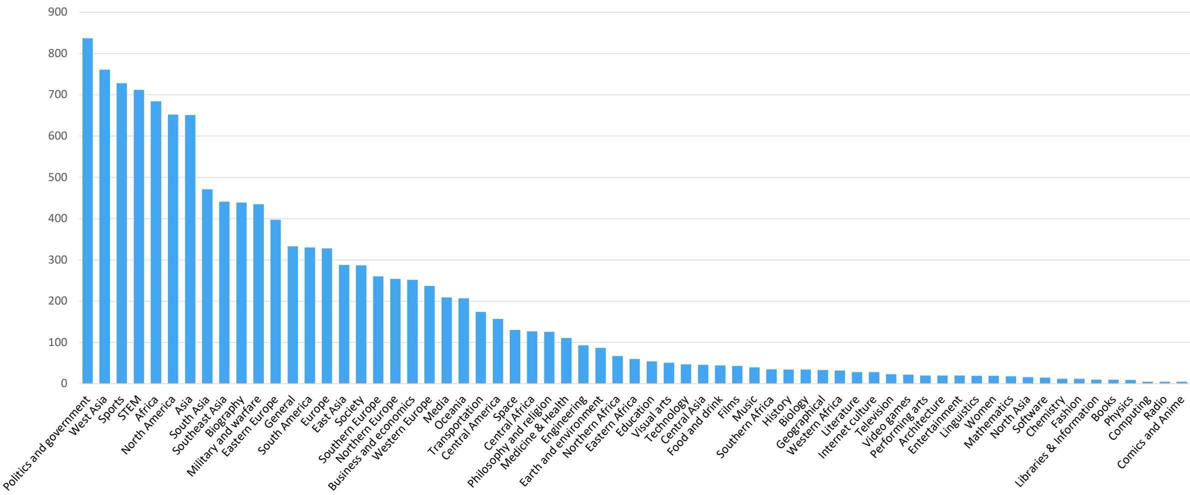

Figure 4: Article Categories Overview