It's About Time: Incorporating Temporality in Retrieval Augmented Language Models

No Author Given

No Institute Given

Abstract. Ensuring that users receive the most relevant and up-to-date information, especially in the presence of multiple versions of web content from different time points remains a critical challenge for information retrieval. In this paper, we propose and evaluate TEMPRALM, a temporally-aware retrieval-augmented language model with few-shot learning capabilities. TEMPRALM considers both the semantic and temporal relevance of retrieved documents in relation to a given query, rather than relying solely on semantic similarity. Our approach demonstrates up to a 74% improvement in performance over the baseline model ATLAS and a 32% improvement over a state-of-the-art LLM with augmented with retrieval. We also introduce a novel automated method for generating ground truth data for RALMs and question-answering (QA) tasks, which we use to create Tablepedia - a temporal question-answering dataset focused on tennis facts. We evaluate TEMPRALM on Tablepedia, demonstrating its effectiveness without requiring model pre-training, document index replacement, or other computationally intensive elements.

Keywords: Information Retrieval · Natural Language Processing · Large Language Models.

1 Introduction

The web is a vast source of real-world knowledge, with much of it in textual form. Moreover, information changes over time, leading to updates to existing documents, or the addition of new documents. This leads to multiple versions of information from various time frames to co-exist and grow over time. Ensuring users access the most relevant, up-to-date content is a key challenge in information retrieval, particularly with the rise of LLM-based question-answering tools like ChatGPT (10). These models typically learn from a static snapshot of web data (11), however, real-world information changes constantly, frequently on a daily, hourly or even real-time basis. Regularly updating LLMs to reflect new data is crucial but demands considerable computational and financial resources.

Interest in Retrieval Augmented Language Models (RALMs) has grown to address evolving information and reduce hallucinations in LLMs. RALMs use an external document corpus such as Wikipedia, stored as an indexed database, allowing for efficient retrieval and updates. For text-based queries, a RALM's

retriever selects relevant documents from its index, which the language model (reader) uses to generate responses. RALMs provide more specific and factual answers in knowledge-intensive tasks compared to stand-alone LLMs (9), and perform well in few-shot training scenarios, achieving good results with as few as 64 examples (7).

While RALMs excel in factual question answering, they typically rely on a document index containing a single version of each document. In many real-world scenarios, however, new information is continually generated without invalidating existing data, leading to multiple document versions. For instance, scientific and medical journals frequently publish new papers that build on previously published knowledge, e.g. new research papers being added to platforms like arXiv almost daily! (1). We show that state-of-the-art LLMs and RALMs struggle with temporality. For instance, GPT-3.5 and ATLAS (7) fail to provide timely answers to frequently changing information, such as Wimbledon winners. Asking "Who won the Wimbledon Men's championship?" on December 31, 2019, yields relevant results since the query aligns with the match date. However, asking the same question a day later, on Jan 1, 2020, fails due to a misalignment between the query's year and the document's year, highlighting these models' inability to "understand" temporal relationships.

In this paper, we introduce TEMPRALM, a temporal retrieval and ranking algorithm that enhances the ATLAS (7) document retriever, improving performance on temporal queries by up to 74% with minimal overhead. We evaluate its transferability in a RALM setting using GPT-3.5, improving its performance with temporally + semantically retrieved passages by 32% in comparison to using only semantically retrieved passages.

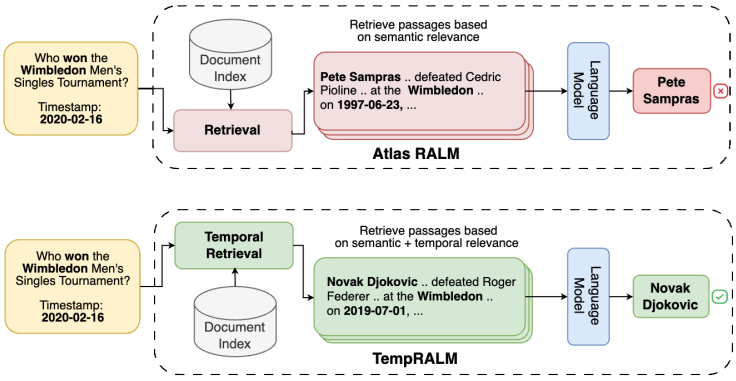

Fig. 1. TempRALM: This figure highlights how TempRALM, unlike RALMs, retrieves documents based on both semantic and temporal relevance to the query.

2 Related Work

Large Language Models (LLMs) (2), (5), and conversational interfaces on these models such as ChatGPT (10) show great promise in information retrieval, particularly in Question-Answering (2). An aspect of question answering that has emerged as a challenge for both RALMs and LLMs are questions whose answers evolve over over time as seen in (16). In (4), the authors examine the challenges posed to LLMs by time-sensitive questions, demonstrating that LLMs pre-trained on a static snapshot are inherently incapable to answer them correctly. However, these models require extensive training datasets, and frequently updating them for time-sensitive questions is impractical.

To alleviate the need of large datasets for training or fine-tuning RALMs, (7) introduce ATLAS, a pre-trained model designed for few-shot learning. It features the Contriever (6), a dense retriever based on BERT (3) trained with contrastive loss, and the T5 language model (12). The authors explore the time sensitive QA issue with ATLAS, showing that replacing the document index with an updated one improves performance over closed-book LLMs. However, this modification is insufficient for question answering in scientific, medical, and legal domains, where updating knowledge often involves adding new documents instead of simply replacing existing ones.

In (15), the authors also explore time-sensitive open domain QA, by enhancing a Dense Passage Retriever (8) with a query containing additional linguistic context, but see no improvement for temporally dependent questions. They note that retrieval based models were able to update some world knowledge after swapping the retrieval corpus and fine-tuning with newer data, however it is not efficient to keep models' responses up-to-date. In TEMPRALM, we propose a method accommodates a document index with multiple document versions and does not require model re-training. This approach allows for the addition of information without needing to replace or invalidate prior content.

3 TEMPRALM Overview

3.1 Semantic and Temporal Retrieval Scores

The core computation of a RALM retriever to assess document relevance for a query involves calculating a semantic similarity score between the query and each indexed document. Specifically, for a query q and a document d, the semantic score $s(q, d)$ is computed as the dot product of their encoder representations:

$$s(q, d) = \langle f_\theta(q), f_\theta(d) \rangle \tag{1}$$

We calculate temporal score $\tau(qt, dt)$, where qt is the timestamp of the query q, and dt is the timestamp of document d. We take the reciprocal of the time difference between qt and dt to ensure smaller differences in time result in a larger score:

$$\tau(qt, dt) = \frac{1}{qt - dt} \tag{2}$$

3.2 Temporally-Augmented Retrieval

In our temporally-augmented retriever, we implement a scoring function that takes into account both the semantic and temporal scores of documents retrieved from the index. The first step in this process is to align the temporal score with the numerical range of the semantic score $s(q, d)$ from equation 1, for which, we take inspiration from z-normalization to shift and scale the temporal score

$$\tau(qt, dt) = \frac{\tau(qt, dt) - \mu_\tau}{\sigma_\tau} \times \sigma_s + \mu_s \tag{3}$$

To incorporate the temporal score shown in equation (3) in the retriever, we define an auxiliary retrieval scoring function $TempRet_t$, which is the sum of the semantic score $s(q, d)$ and the temporal score $\tau(qt, dt)$:

$$TempRet_t(q, d, qt, dt) = s(q, d) + \tau(qt, dt) \tag{4}$$

In addition, we mask passages that have a timestamp later than the query timestamp. We re-write the retrieval computation as follows:

$$TempRet_t(q, d, qt, dt) = \{\, s\ (q, d) + \tau(qt, dt)\, if qt \geq dt - \infty, otherwise \tag{5}$$

Finally, to ensure a comprehensive coverage of relevant passages, we follow the paradigm of two-stage retrieval, where we retrieve candidates using semantic scores and re-rank them using them $TempRet_t$ scores.

4 Tablepedia Overview

We introduce TablePedia, a method that converts tabular data into passages for RALM index. Each row of the table is transformed into a natural language passage, with the associated timestamp in YYYY-MM-DD format, included as metadata. This process is depicted in Figure 2.

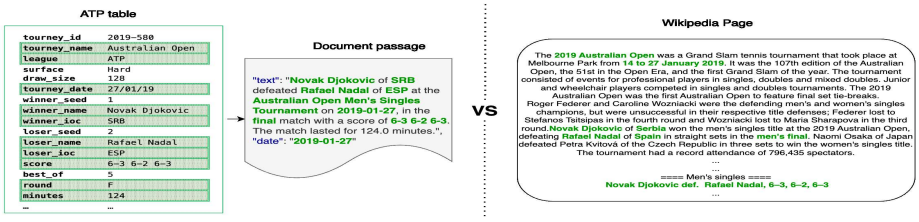

Fig. 2. Example of tabular data to passage conversion.

4.1 Datasets

We create 2 custom datasets with TablePedia technique, to demonstrate tranferability of our methodology, and use them to evaluate TempRALM method on ATLAS and GPT-3.5-RALM

ATP Tennis dataset This TablePedia generated dataset span tournaments from 1978 to 2019, yielding to 50,000 passages, covering matches in Mens' and Women's tennis grand slams, sample passage shown in Fig. 2.

Oscar movies dataset This dataset was created in a similar format as the tennis dataset, which contains a knowledge bank of Oscars nominees and winners from 1922-2019, covering several award categories such as Directing, Best Actor in a supporting role, etc. and contains 10,000 passages.

4.2 Training Set

Focusing on temporally aware question answering, we use an input data format of (query, timestamp, answer) for our model. We generate query-answer pairs for each passage, attaching a timestamp to each query. This timestamp reflects when the query is asked, and the answer corresponds to what is valid at that specific time.

Few-Shot Training Set for ATLAS We create 3 non-overlapping few-shot training sets of size 32, 64 and 128 data points each.

Few-Shot Prompts for GPT-3.5 In our GPT-3.5 experiments, we evaluated the model's performance using zero-shot (0x) and four-shot (4x) prompt configurations, paired with retrieved passages, We selected exemplars for the prompt from few-shot training set of 32.

Query	Timestamp	Answer	Corresponding Passage
Who won the Australian Open women's Singles final	03/10/19	Naomi Osaka	Naomi Osaka of JPN defeated Petra Kvitova of CZE at the Australian Open Women's Singles Tournament on 2019-01-14, in the final match with a score of 7-6(2) 5-7 6-4. The match lasted for 147.0 minutes.
Who was runner-up in the Australian Open women's Singles final	10/15/19	Petra Kvitova	
Who played in the Australian Open Women's Singles final?	01/01/20	Naomi Osaka and Petra Kvitova	
What was the score in Australian Open Women's Singles final?	01/01/20	7-6(2) 5-7 6-4	

Fig. 3. Questions-Answer pairs generated on a passage

4.3 Test sets

We created two 128-item evaluation test sets, TPQ-2019 and TPQ-2020, short for Temporal Proximity Query sets. Each set uses ground truth triplets focused on data from 2019 and 2020 to ensure a clear separation from the training timeline. Both sets contain identical questions but differ in year-specific timestamps, allowing us to simulate two temporal contexts. Correct answers require the model to demonstrate temporal awareness beyond simple pattern matching by recognizing date proximity.

5 Methodology

In this section, we outline our baseline comparison approach, evaluation metrics, and the experiments we conducted. Our dataset is described in detail in Section 4.

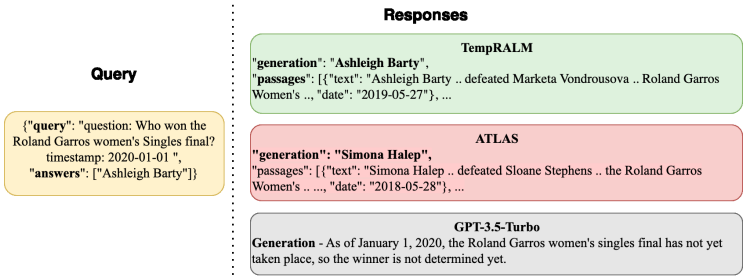

Fig. 4. Examples of responses generated for a TPQ-2020 question by GPT-3.5, ATLAS, and TEMPRALM

5.1 State-Of-The-Art Commercial LLM Baseline

Our focus in this paper is on free open-source models that can be trained on reasonable compute infrastructure such as university high compute cluster, where a single GPU may be available for us at a time. However, to get a sense of the capabilities of commercial state-of-the-art model when it comes to factual questions with temporal characteristics, we conducted experiments with the GPT-3.5 Turbo model class (13), and more specifically gpt-3.5-turbo-0613 model. The GPT-3.5-Turbo model is used in ChatGPT-3.5 (10) and is trained with data up to September 2021. As a reference, all of the queries in our evaluation set require information from 2020 or earlier and thus GPT-3.5 has been trained on the data required to answer them.

We note that ATLAS and TEMPRALM have 770M parameters for the LLM, and 100M parameters for the retriever, whereas GPT-3.5 Turbo, has 175B+ model parameters (14), which makes it nearly 1000x larger in size.

5.2 RALM Baseline comparison

Since TEMPRALM is an augmentation of ATLAS, we use ATLAS as our primary baseline for comparison. For our ATLAS experiments, we use the standard ATLAS-large configuration (7) configuration. We use the same model size for the TEMPRALM experiments as well. We conducted an extensive set of experiments to chose parameters for TEMPRALM and ATLAS with a range of values across their hyper-parameters, including *topk* documents to retrieve for every question, number of training steps, retriever and language model learning rates, and sampling temperatures.

We conduct a series of experiments to test the temporal awareness of ATLAS with few-shot settings, using our time-sensitive dataset that contains text passages that refer to tennis tournament details at different time points, described in Section 4.

5.3 Prompt based Few-shot Temporal Learning with GPT-3.5 RAG

In order to broaden the scope of our analysis, we conducted additional experiments using an alternative baseline LLM, GPT-turbo-3.5 (10), within the

retrieval-augmented generation (RAG) framework. Referred to here as GPT-turbo-3.5-RAG, this model replaces the Atlas LLM in the RAG pipeline. The experiments involved four distinct settings to evaluate GPT-3.5's capabilities: (1) prompting the model with a question in a closed-book format, (2) prompting the model with a question and a selection of passages from which it chooses an answer, (3) providing a question and passages retrieved using TempRALM, and (4) supplying a question, passages, and four illustrative examples consisting of (question, passages, correct answer) pairs to guide the model in identifying the relevant passage and producing an accurate response.

5.4 Evaluation Metrics

We evaluate the effectiveness of our system on time-specific question answering test sets. We assume a few-shot training set of (query, timestamp, answer) triplets,
$Q_{train} = \{(q_1, qt_1, a_1), (q_2, qt_2, a_2), \ldots (q_N, qt_N, a_N\}$ and evaluate on a test set of held out queries, timestamps, and answers. Our evaluation metric is exact-match between the predicted and ground-truth answer for each data point in the test set.

5.5 Experiments

We designed our experiments in the following order to thoroughly evaluate TEMPRALM's effectiveness and improvements in handling time-sensitive information

- **LLM Baseline – chatGPT:** In this experiment, we use GPT-3.5 Turbo (chatGPT) in a closed-book setup to evaluate how well it handles time-sensitive questions without external retrieval support. Using exact match scoring on the TPQ-2019 and TPQ-2020 datasets, we assess GPT-3.5's accuracy in answering questions that rely on specific timing. This closed-book performance establishes a baseline for comparison with later retrieval-augmented setups and TEMPRALM's improvements.
- **RALM baseline ATLAS**: In this experiment, we used ATLAS as the baseline RALM to effectively compare how TEMPRALM improves handling time-sensitive questions. We conducted the experiments in two parts to gain a deeper understanding of performance:
  - Model size (base vs large): We tested both ATLAS-base (Reader: 220M parameters / Retriever: 110M parameters) and ATLAS-large (Reader: 770M parameters / Retriever: 110M parameters) to see how model size impacts performance on temporally relevant queries. By comparing ATLAS-base and ATLAS-large on the TPQ-2019 and TPQ-2020 datasets, we aimed to determine if the larger model could better handle time-dependent questions and gain more from temporal augmentation in TEMPRALM. These tests used a 64-shot training setup for both model sizes.

  - Retriever recall metrics: To assess how well the ATLAS retriever ranks relevant documents, we used Recall@1 and Recall@5 metrics, which indicate how often the correct passage appears as the top result (Recall@1) or within the top five results (Recall@5). This approach provided insight into each model's retrieval accuracy. For this part, we ran 64-shot training on both ATLAS-large and TEMPRALM, focusing especially on TPQ-2020, where temporal information is essential for accuracy.
- **GPT-3.5 RALM vs GpT-3.5 TempRALM**: To evaluate TEMPRALM's effectiveness on other language models and compare it with other RALM techniques, we designed an experiment using GPT-3.5 within both standard RALM and TEMPRALM frameworks. This setup allowed us to analyze how temporal augmentation impacts GPT-3.5's performance on time-sensitive tasks.
  - We experimented with small (1,536 tokens) and large (3,072 tokens) embeddings to understand the effect of embedding size on retrieval accuracy. This comparison was essential to see if larger embeddings provide better retrieval quality in time-sensitive contexts. By adjusting embedding size, we aimed to evaluate how well each configuration captures temporal nuances in the retrieved data, specifically within challenging scenarios like TPQ-2020, where precision in recent context is critical.
  - We tested zero-shot and four-shot prompting strategies to measure how few-shot prompting affects GPT-3.5's ability to handle questions that rely on temporal context. Zero-shot prompting requires the model to generate answers without any examples, while four-shot prompting provides a few illustrative examples to guide the model. This experiment was designed to examine if additional prompt examples improve TEMPRALM's response accuracy on temporally dependent questions, allowing us to fine-tune the model's ability to handle nuanced, time-restricted queries.
  - To assess TEMPRALM's generalizability, we introduced a new dataset, Tablepedia-Oscars, extending testing beyond tennis data. By incorporating a dataset on Oscars nominees and winners, we explored how well TEMPRALM adapts to different types of time-sensitive information and domains. This cross-domain testing was crucial to understanding TEMPRALM's flexibility and effectiveness beyond a single use case, verifying its utility in handling temporally restricted queries across varied datasets.

We summarize the experiments we conducted below and report results in Section **??**.

- Experiment 1: For our LLM baseline experiment, we evaluate the performance of GPT-3.5 on our evaluation sets, TPQ-2019 and TPQ-2020, using an exact match metric.
- Experiment 2: For our RALM baseline experiment, we compare the performance of TempRALM to Atlas-large on the TPQ-2019 and TPQ-2020 test

sets in 32, 64, and 128 few-shot training scenarios, using an exact match metric.
- Experiment 3: We evaluate the retrievers of both TEMPRALM and Atlas using a recall@1 and recall@5 metrics, which compute the number of times the correct passage is in the top 1 or top 5 results returned by the retriever respectively.
- Experiment 4: We explore the impact of model size on performance by comparing the performance of TempRALM-base vs. TempRALM-large, and the baseline case of Atlas-base vs. Atlas-large.

All of our experiments are conducted on a single NVIDIA A100-PCIE-40GB GPU.

6 Results

		Tennis Dataset			Oscar Dataset		
		32-shot	64-shot	128-shot	32-shot	64-shot	128-shot
3*TPQ-2019	ATLAS	64.84	74.68	76.88	82.14	85.71	92.86
	TempRALM-ATLAS	71.72	75.78	77.8	92.86	92.86	92.86
	Δ	**6.88**	**1.1**	**0.92**	**10.72**	**7.15**	**0**
3*TPQ-2020	ATLAS	37.65	40.93	41.72	32.14	53.57	64.29
	TempRALM-ATLAS	55.93	68.44	72.65	82.14	85.71	92.86
	Δ	**18.28**	**27.51**	**30.93**	**50**	**32.14**	**28.57**

Table 1. Comparing Temporal Proximity with Time based token matching: We evaluate the retrieval performance of two query sets - TPQ-2019 and TPQ-2020 - which contain identical questions about an event that occurred in 2019 but have timestamps from two consecutive years, to simulate the time-frame in which the queries are made. This table presents the performance comparison between the Atlas-large model and TempRALM. Both models were fine-tuned on a limited training sets comprising 32, 64, and 128 time-suffixed question-answer pairs. A robust model is expected to accurately address queries based on their temporal closeness to passages relevant to the event, rather than relying solely on matching year tokens within the queries and passages. Our model demonstrates strong performance across both query sets.

We report the results of our experiments next.

6.1 State-Of-The-Art Commercial LLM Baseline Results

The results of our GPT-3.5 experiments are shown in Table 2. We find that GPT-3.5's exact match accuracy is **68.75% for TPQ-2019 and 65.62% for TPQ2020**, and is lower than that of TEMPRALM with 64 or more few-shot training examples (Table 1). This is notable especially since GPT-3.5 is 1000 times larger in size than TEMPRALM and ATLAS (Section 5.1). As we mentioned in Section 5, GPT-3.5 is trained with data up to September 2021. As a

reference, all of the queries in our evaluation set require information from 2020 or earlier and thus GPT-3.5 has been trained on the data required to answer them.

There are several classes of mistakes that we observed GPT-3.5, which we illustrate below using examples from the TPQ-2019 experiment:

Does not understand the concept of current winner (current reigning champion):

- Question: Who won the Australian Open men's Singles final as of 2020-01-01?
- GPT-3.5 Answer: As of 2020-01-01, the 2020 Australian Open Men's Singles final has not taken place yet, so the winner is not known.
- The correct response should have been "Novak Djokovic".

Confuses both the timing of the event and the finalist status (winner vs. runner-up):

- Question: Who was runner-up in the Australian Open women's Singles final as of 2020-01-01?
- GPT-3.5 Answer: As of January 1, 2020, Sofia Kenin of the United States was the runner-up in the Australian Open women's singles final.
- Sofia Kenin was the winner of the 2020 Australian Open women's singles final, not the runner-up in the Australian Open as of 2020-01-01. The correct response should have been "Petra Kvitova"

GPT-3.5 is confused about which date comes first - September 2021 or January 1, 2020

- Question: What was the score in Australian Open Women's Singles final as of 2020-01-01?
- GPT-3.5 Answer: I'm sorry, but as of my knowledge cutoff date in September 2021, I cannot provide the specific score of the Australian Open Women's Singles final on January 1, 2020. However, I can inform you that the Australian Open 2020 Women's Singles final was held on February 1, 2020, and Sofia Kenin defeated Garbiñe Muguruza with a score of 4-6, 6-2, 6-2 to win the title.

Test Set	TPQ-2019	TPQ-2020
GPT-3.5 Turbo	68.75	65.62

Table 2. GPT-3.5-turbo Evaluation Results on TPQ-2019 and TPQ-2020.

6.2 RALM Baseline Comparison Results

We compared TEMPRALM and ATLAS-large on the TPQ-2019 and TPQ-2020 test sets using the exact match metric (Table 1), with each experiment run 5 times and results averaged. When the query year matches the event year in the text, ATLAS performs comparably to TEMPRALM, as the semantic score aligns the dates.

However, in cases where the query and passage years differ (TPQ-2020), TEMPRALM outperforms ATLAS by 49% in the 32-shot, 67% in the 64-shot, and 74% in the 128-shot experiments. This highlights the importance of temporal augmentation, as it captures time proximity between mismatched timestamps, which semantic scores alone miss.

Our results also show that as few-shot training examples increase, TEMPRALM's impact grows significantly. Consistent with (7), we observe that performance stabilizes at 64 examples and improves with 128 examples. Exact match performance is detailed in Table 1

Retriever performance analysis We observed instances where TempRALM successfully retrieves the gold passage (which contains the correct answer), but the answer generated by the LLM is wrong, indicating that the problem in those examples lies with the LLM.

	TPQ-2019		TPQ-2020	
Model	Recall@1	Recall@5	Recall@1	Recall@5
Atlas-Large	0.54	0.63	0.24	0.63
TempRALM	0.63	0.75	0.64	0.75

Table 3. Retriever Recall Metrics: We calculate the recall in 64-shot training example setting. We choose an experiment closest to the average exact match of all our 64-shot Atlas and TempRALM experiments.

To quantify this behavior, we calculated how often the retriever ranked the gold passage as the top result (recall@1) and within the top 5 results (recall@5) for both TempRALM and Atlas-large. Table 3 shows that temporal retrieval improves both Recall@1 and Recall@5, with the most notable improvement in Recall@1 for TPQ-2020, where TempRALM outperforms Atlas-large by 165%. These findings highlight Atlas-large's limitations with temporal data and TempRALM's strength in handling it effectively.

Impact of Model Size on Performance: Large vs Base Model We evaluated the Atlas large model (770M reader / 110M retriever parameters) and compared its performance to the Base model (220M reader / 110M retriever). Results show that temporal retrieval improves performance across different model sizes.

Table 4 presents the Exact Match results in a 64-shot experiment for both Atlas and TempRALM, demonstrating the effectiveness of TempRALM's temporal retrieval.

(r)1-3	Test Set Exact Match	
(lr)2-3 Model Size	TPQ-2019	TPQ-2020
(r)1-3 Atlas Base	42.97	28.13
TempRALM Base	56.25	35.94
(r)1-3 Atlas Large	74.68	40.93
TempRALM Large	75.78	68.44
(r)1-3		

Table 4. Model Size Performance: We compare the Exact Match in 64-shot training example setting between Atlas and TempRALM across base and large models.

6.3 GPT-3.5 augmented with Retrieval vs GPT-3.5 augmented with TempRALM

	Tennis		Oscars	
GPT experiment Setup	TPQ-2019	TPQ-2020	TPQ-2019	TPQ-2020
GPT-3.5 - closed book	68.75	65.62	57.14	17.8
GPT-3.5 - RAG with small-embeddings	34.38	12.5	–	–
GPT-3.5 - TEMPRALM with small-embeddings	59.38	43.75	–	–
GPT-3.5 - RAG with large-embeddings	43.8	43.75	65.38	57.69
GPT-3.5 - TEMPRALM with large-embeddings	75	56.25	73.07	76.92
GPT-3.5 - RAG + 4-shot with large embeddings	90.63	50	73.07	42.03
GPT-3.5 - TEMPRALM + 4-shot with large embeddings	93.75	65.63	84.61	76.92

Table 5. Performance comparison of GPT-3.5 in closed-book, retrieval-augmented (RAG), and TEMPRALM-enhanced setups across TPQ-2019 and TPQ-2020 datasets for tennis and Oscars, showing the impact of temporal augmentation, embedding size, and prompting strategies on accuracy.

The experiment compares GPT-3.5 in a retrieval-augmented setup (RAG) with TEMPRALM-augmented GPT-3.5 on time-sensitive datasets, with results shown in 5. Findings reveal that TEMPRALM-enhanced GPT-3.5 outperforms standard RAG-GPT-3.5, particularly on the TPQ-2020 dataset, highlighting the effectiveness of temporal awareness for handling time-specific queries.

TEMPRALM shows a 34-point improvement in the Oscars dataset compared to 16 points in the tennis dataset, indicating its advantage in domains with distinct yearly events. In contrast, the recurring nature of tennis winners limits temporal distinctiveness, making TEMPRALM's temporal adjustments particularly impactful for the Oscars dataset.

Closed-book GPT-3.5 struggles on TPQ-2020 in the Oscars dataset (17.8% accuracy) compared to tennis (65.62%), likely due to the Oscars' added temporal complexity. The retrieval-augmented model helps address these nuances by contextualizing the timeline.

Error analysis shows GPT-3.5 RAG still generates incorrect answers in up to 35% of cases even with correct passages, highlighting language model limitations rather than retrieval issues. TEMPRALM mitigates this by accurately contextualizing data.

In conclusion, TEMPRALM enhances GPT-3.5's handling of temporally sensitive queries, particularly in complex domains, demonstrating its potential for broader applications in time-specific retrieval tasks.

7 Conclusions and Future Work

In this study, we introduced and evaluated TempRALM, a Retriever Augmented Language Model (RALM) augmented with temporal awareness, and a novel automated method for generating factual documents from tables, which does not require manual validation, along with Tablepedia, a dataset generated by our method, which we use for our model evaluation. Unlike conventional RALM approaches that rely solely on semantic similarity, TempRALM considers both semantic and temporal relevance when selecting documents to pass to its Large Language Model (LLM) in response to a given query. Our results indicate an improvement in performance of up to 74% compared to the Atlas-large model, even when multiple versions of documents (from different time points) are present in the document index. Notably, we achieve this without the need for model pretraining, replacing the document index with an updated index, or adding any of other computationally intensive elements. We plan to explore a number of avenues for building on the work presented in this paper, such as implementing and evaluating different learning strategies for the parameters of our temporal relevance function, and exploring the interplay between the retriever and LLM. As documents get more complex, another avenue of our research is to consider implicit timestamps present in the document along with the file-level timestamps, as that will give a more comprehensive sense of temporality. Temporality can also show promise in multi-modal systems where images and videos are retrived based on their upload timestamps. Furthermore, we plan to explore the use of our temporal retrieval approach in other tasks such as fact checking, recommender systems, and retrieval augmented dialog agents.

Acknowledgments. A bold run-in heading in small font size at the end of the paper is used for general acknowledgments, for example: This study was funded by X (grant number Y).

Disclosure of Interests. It is now necessary to declare any competing interests or to specifically state that the authors have no competing interests. Please place the statement with a bold run-in heading in small font size beneath the (optional)

acknowledgments[1], for example: The authors have no competing interests to declare that are relevant to the content of this article. Or: Author A has received research grants from Company W. Author B has received a speaker honorarium from Company X and owns stock in Company Y. Author C is a member of committee Z.

[1] If EquinOCS, our proceedings submission system, is used, then the disclaimer can be provided directly in the system.

Bibliography

[1] Arxiv: Arxiv (1991), https://arxiv.org

[2] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)

[3] Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018), http://arxiv.org/abs/1810.04805

[4] Dhingra, B., Cole, J.R., Eisenschlos, J.M., Gillick, D., Eisenstein, J., Cohen, W.W.: Time-aware language models as temporal knowledge bases. CoRR **abs/2106.15110** (2021), https://arxiv.org/abs/2106.15110

[5] Google, R.A., Dai, A.M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., Chu, E., Clark, J.H., Shafey, L.E., Huang, Y., Meier-Hellstern, K., Mishra, G., Moreira, E., Omernick, M., Robinson, K., Ruder, S., Tay, Y., Xiao, K., Xu, Y., Zhang, Y., Abrego, G.H., Ahn, J., Austin, J., Barham, P., Botha, J., Bradbury, J., Brahma, S., Brooks, K., Catasta, M., Cheng, Y., Cherry, C., Choquette-Choo, C.A., Chowdhery, A., Crepy, C., Dave, S., Dehghani, M., Dev, S., Devlin, J., Díaz, M., Du, N., Dyer, E., Feinberg, V., Feng, F., Fienber, V., Freitag, M., Garcia, X., Gehrmann, S., Gonzalez, L., Gur-Ari, G., Hand, S., Hashemi, H., Hou, L., Howland, J., Hu, A., Hui, J., Hurwitz, J., Isard, M., Ittycheriah, A., Jagielski, M., Jia, W., Kenealy, K., Krikun, M., Kudugunta, S., Lan, C., Lee, K., Lee, B., Li, E., Li, M., Li, W., Li, Y., Li, J., Lim, H., Lin, H., Liu, Z., Liu, F., Maggioni, M., Mahendru, A., Maynez, J., Misra, V., Moussalem, M., Nado, Z., Nham, J., Ni, E., Nystrom, A., Parrish, A., Pellat, M., Polacek, M., Polozov, A., Pope, R., Qiao, S., Reif, E., Richter, B., Riley, P., Ros, A.C., Roy, A., Saeta, B., Samuel, R., Shelby, R., Slone, A., Smilkov, D., So, D.R., Sohn, D., Tokumine, S., Valter, D., Vasudevan, V., Vodrahalli, K., Wang, X., Wang, P., Wang, Z., Wang, T., Wieting, J., Wu, Y., Xu, K., Xu, Y., Xue, L., Yin, P., Yu, J., Zhang, Q., Zheng, S., Zheng, C., Zhou, W., Zhou, D., Petrov, S., Wu, Y.: Palm 2 technical report (2023)

[6] Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., Grave, E.: Towards unsupervised dense information retrieval with contrastive learning. CoRR **abs/2112.09118** (2021), https://arxiv.org/abs/2112.09118

[7] Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., Grave, E.: Atlas: Few-shot learning with retrieval augmented language models (2022)

[8] Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., tau Yih, W.: Dense passage retrieval for open-domain question answering (2020)

[9] Lewis, P.S.H., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive NLP tasks. CoRR **abs/2005.11401** (2020), https://arxiv.org/abs/2005.11401

[10] Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., et al.: Summary of chatgpt/gpt-4 research and perspective towards the future of large language models. arXiv preprint arXiv:2304.01852 (2023)

[11] Petroni, F., Rocktäschel, T., Lewis, P.S.H., Bakhtin, A., Wu, Y., Miller, A.H., Riedel, S.: Language models as knowledge bases? CoRR **abs/1909.01066** (2019), http://arxiv.org/abs/1909.01066

[12] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. CoRR **abs/1910.10683** (2019), http://arxiv.org/abs/1910.10683

[13] gpt turbo: gpt-turbo https://platform.openai.com/docs/models/gpt-3-5-turbo

[14] Ye, J., Chen, X., Xu, N., Zu, C., Shao, Z., Liu, S., Cui, Y., Zhou, Z., Gong, C., Shen, Y., Zhou, J., Chen, S., Gui, T., Zhang, Q., Huang, X.: A comprehensive capability analysis of gpt-3 and gpt-3.5 series models (2023)

[15] Zhang, M.J.Q., Choi, E.: Situatedqa: Incorporating extra-linguistic contexts into qa (2021)

[16] Zhao, W.X., Liu, J., Ren, R., Wen, J.R.: Dense text retrieval based on pretrained language models: A survey (2022)