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

Anoushka Gade
Computer Engineering Department
San Jose State University
San Jose, USA
anoushka.gade@gmail.com

Jorjeta G. Jetcheva
Computer Engineering Department
San Jose State University
San Jose, USA
jorjeta.jetcheva@sjsu.edu

Hardi Trivedi
Computer Engineering Department
San Jose State University
San Jose, USA
hardi.trivedi@sjsu.edu

***Abstract*—In this paper, we propose and evaluate TempRALM, a temporally aware retrieval-augmented language model with few-shot learning capabilities, which considers both the semantic and temporal relevance of retrieved documents in relation to a given query, rather than relying on semantic similarity alone. Our approach demonstrates up to 74% improvement in performance over the baseline state-of-the-art retrieval-augmented language model ATLAS, and 32% improvement over a state-of-the-art commercial large language model augmented with retrieval. TempRALM achieves these improvements without requiring model pre-training, document index replacement, or other computationally intensive operations. Additionally, we introduce and evaluate TablePedia, a novel automated method for generating ground truth data for retrieval-augmented language models and temporal question-answering.**

***Index Terms*—Information Retrieval, Temporality, Retrieval Augmented Language Models**

I. Introduction

Large Language Models (LLMs) [1], [2] and conversational interfaces based on these models, such as ChatGPT [3], show great promise for information retrieval [4], particularly in question-answering (QA) [1]. However, these models are pre-trained on a static snapshot of text data [5], whereas, real-world information changes constantly, frequently on a daily, hourly or even real-time basis. Retrieval-Augmented Language Models (RALMs) [6] are a popular approach for addressing changing information and also grounding LLMs to mitigate hallucinations [7]. RALMs typically consist of an LLM, a retriever, and an external document corpus such as Wikipedia, stored as an index, which allows for efficient retrieval and updates. For a given text-based query, RALM uses its retriever component to obtain a ranked list of the most relevant documents from the document index (e.g., the $top\text{-}k$ documents) and passes them to its language model component (also known as a reader), which uses the ranked documents as context when it generates a response to the query. This approach is referred to as Retriever-Augmented Generation (RAG).

In [8], the authors examine the challenges posed to LLMs by time-sensitive questions, demonstrating that LLMs pre-trained on a static snapshot are inherently incapable of answering them correctly. At the same time, regularly updating LLMs to reflect new data demands considerable computational and financial resources [9] and is not feasible. To alleviate this need of re-training or fine-tuning LLMs and RALMs, [10] introduce ATLAS, a pre-trained model designed for few-shot learning. The authors present an approach for handling time-sensitive queries in ATLAS, showing that replacing the entire document index with one that contains only the most recent version of each document improves performance over closed-book LLMs (which only rely on knowledge extracted during their pre-training). However, this modification is insufficient for question answering in scientific, medical, or legal domains, where updating knowledge often involves adding new documents rather than simply replacing existing ones.

In this paper, we propose and evaluate TempRALM, a temporally-aware augmentation for retrieval-augmented language models, which considers both the semantic and temporal relevance of retrieved documents in relation to a given query, and enables the most recent version of information to be retrieved without requiring model retraining, document index replacement, or detection and removal of obsolete information from documents (Fig. 1). We evaluate TempRALM by augmenting the state-of-the-art open-source RALM ATLAS [10] and commercial LLM GPT-3.5 [11] in a RALM setting, and show that it improves their performance on temporal queries by 74% and 32%, respectively.

In addition, we introduce a novel automated method for generating datasets for temporally aware factual question answering from commonly available tables with summaries of temporal facts such as the ATP/WTA Men's and Women's Singles Tournament Tennis data table [12]. Our method, which we refer to as Tablepedia, is fully automated and results in ground truth passages that do not require any manual validation. In contrast, extracting facts through semantic analysis, is error-prone and requires validation to ensure ground truth accuracy. Using our dataset generation method, we create a 50,000+ document dataset focused on tennis-related facts, as well as a 10,000+ document Oscars dataset, and use them for our performance evaluation. Due to the large number of table and spreadsheet-based summaries of common information, including information with temporal properties, we believe that our method has broad applicability beyond the use cases we explore in this work.

The rest of this paper is organized as follows. Section II

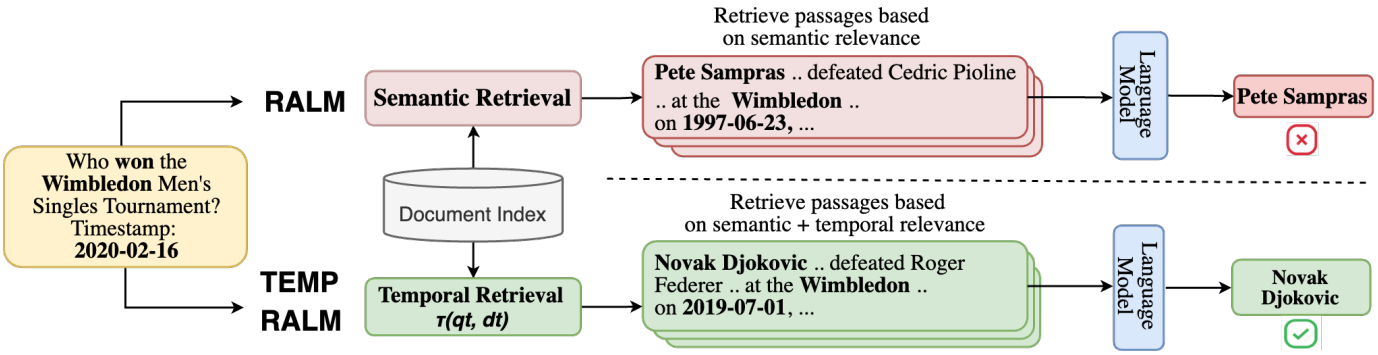

Fig. 1. Example of query answering process in TempRALM vs. standard RALMs.

overviews related work. We introduce TempRALM and Tablepedia in Sections III and IV respectively. We report on the results from our evaluation in Section V and conclude the paper with a summary in Section VI, and next steps in Section VII.

II. Related Work

To alleviate the need of large datasets for training or fine-tuning RALMs, [10] propose ATLAS, a pre-trained model with few-shot learning capabilities. It consists of a pre-trained retriever, [13], which is a dense retriever based on the BERT [14] architecture, trained using contrastive loss, as well as a language model, T5 [15], trained to perform a range of sequence-to-sequence tasks, e.g., summarization, translation, and question answering. Additionally, both models are fine-tuned jointly as a Fusion-in-Decoder [16] on common-crawl data using Wikipedia as a document index. The authors showed that ATLAS performs well out-of-the-box and can be adapted to different tasks such as question-answering and fact-checking with as few as 64 few-shot examples.

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 [17]. For example, the answer to "Who is the President of the United States?" will have a different answer every 4 years or so.

In [8], the authors explore the challenges such questions pose to LLMs, and show how LLMs pre-trained on a single snapshot of web data at a given point in time are inherently incapable of answering them correctly. In their work in [10], the authors investigate this problem in the context of RALMs and show that RALMs can handle these questions better than pure LLMs, by replacing the document index with an up-to-date document index. However, this modification is not sufficient to address the answer of questions in domains such as scientific, or medical publications, where the information is added gradually over time. The older information usually does not get invalidated by the new additions.

In [18], the authors also explore time-sensitive open domain QA, where they provide a Dense Passage Retriever [19] with a query modified with additional linguistic context, but see no improvement for temporally-dependent questions. They also found that retrieval based models were able to update some world knowledge after swapping the retrieval corpus and fine-tuning with newer data. However, this method is not an efficient approach for keeping model responses up-to-date. It is similar to the approach used in ATLAS, where the authors use a static snapshot of Wikipedia, but swap the document index based on the temporal context, which is not feasible in scenarios where there are frequent updates in the data.

In [20], the authors collect temporally evolving questions by sourcing them from news websites and propose a solution utilizing a RAG model with a wiki snapshot to answer the questions. One limiting factor in this approach is that they use a static document index to address dynamically evolving questions. In our approach, we handle multiple versions of each document in order to accommodate cases where information is added, without needing to swap out or invalidate prior information.

In [21], the authors introduce the TemporalWIKI dataset which utilizes the difference between consecutive snapshots of English-language Wikipedia for training and evaluation, respectively. However, the test sets are in RDF format (Subject, Relation, Object) also referred to as TFWIKI-probes. Converting these triplets into QA pairs for our approach would necessitate manual annotation of the queries. Additionally, the dataset spans a timeframe of 4 months, which is not sufficient for our experiments.

We build on the the work by [10] by proposing and evaluating, TempRALM, a novel temporally augmented RALM with few-shot learning capabilities, which can accommodate a document index which contains multiple versions of documents, and does not require model re-training. In addition, we introduce a novel automated framework for generating a RAG document index from tabular data.

III. TempRALM Overview

The core computation of an RALM retriever involves assessing the relevance of a document for a query based on calculating a semantic similarity score between the query and each document in a document index. After calculating the scores with each document, the $top\text{-}k$ documents that are most relevant to the query are considered. For a query q

and a document d, the semantic score $s(q, d)$ is computed as the dot product of their encoder representations, which are obtained independently by encoding them using an encoder f_θ, parameterized by θ, the set of trainable parameters of the encoder [13]:

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

We add an additional component to the similarity score computation which captures the time relevance of the documents by calculating a temporal score $\tau(qt, dt)$, where qt is the timestamp of the query q, and dt is the timestamp of document d. By taking the reciprocal of the time difference between qt and dt, we ensure that smaller differences in time result in a higher score, in order to simulate temporal closeness of a document to the query:

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

In our temporally-augmented retriever, we implement a scoring function that takes into account both the semantic and temporal scores of retrieved documents. To align the temporal score with the semantic score, we take inspiration from z-normalization [22] to shift and scale the temporal score to be in range of the semantic 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 a retrieval scoring function $TempRet_t$, which is the sum of the semantic score $s(q, d)$ and the temporal score $\tau(qt, dt)$. The scoring function $TempRet_t$ gives equal weightage to semantic and temporal scores, to ensure that temporality does not outweigh the semantic similarity of the document to the query:

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

In addition, we mask the retrieval score for retrieved passages that have time stamps that are later than the query timestamp (denoting passages that refer to future information relative to the time of the event specified in the query), to completely eliminate these passages from being considered while ranking the top-k. We re-write the retrieval computation as follows:

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

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 $TempRet_t$. We retrieve 120 documents (vs. 20 used by ATLAS as a default) which we found to work well across a range of settings during our hyperparameter tuning experiments. We refer to this as over-retrieval since we retrieve more documents than are typically retrieved by a standard retriever. From this over-retrieved set, the top-k documents with highest $TempRet_t$ scores are passed as input to the language model. We conduct an extensive set of experiments to find the optimal number of documents to over-retrieve. It is important to note that our extensions do not require re-training any part of the RALM.

We illustrate the difference between semantic retrieval and traditional retrieval in Figure 1.

IV. Tablepedia Overview

To evaluate the effectiveness of our temporally-augmented RALM model, we designed a retrieval task based on use cases where information evolves over time. Specifically, we chose two use cases from different domains in order to illustrate the broad applicability of our approach – tennis grand slam data and Oscars awards data. Both use cases are characterized by factual questions with time-dependent answers. By selecting a well-defined use case with structured facts, we aim to gain insights into the challenges LLMs and RALMs face when handling temporal data. For example, there are four major tournaments each year, and the answers to related queries depend on the timing of the query relative to the tournament.

We initially started out planning to use Wikipedia [21] as a document index, as it is commonly used in knowledge intensive tasks. However, the structural complexity of Wikipedia pages poses challenges to associating events with their corresponding dates due to the presence of extensive supplementary text beyond the core factual content, as illustrated in Figure 2. This would require the alternative approach of extracting facts by using semantic parsing which necessitates extensive manual validation due to limitations related to correctly extracting factual information from text.

To generate accurate answers to questions related to our tennis use case, we began exploring a straightforward method for extracting facts from online sources while preserving their correctness. After crawling all relevant Wikipedia pages and parsing documents related to tennis Grand Slams, we discovered that the ATP/WTA Men's and Women's Singles Tournament Tennis data table [12] offers comprehensive, up-to-date details for each event. It includes information such as match participants, winners, game dates and locations, and scores, as shown in Figure 2. Tables are commonly used to summarize event details and can therefore serve as a reliable source for curating factual answers across a wide range of domains.

The remainder of this section outlines the creation of the document index, which we named Tablepedia data index, along with the corresponding training and evaluation data used in our study.

A. Tablepedia Document Index Generation

We describe our methodology in detail in the context of the tennis use case. We started out by filtering the ATP/WTA table by the following 4 grand slams: Australian Open, Roland Garros, Wimbledon and US Open. We then converted each row of the table into a passage of (natural language) text using a

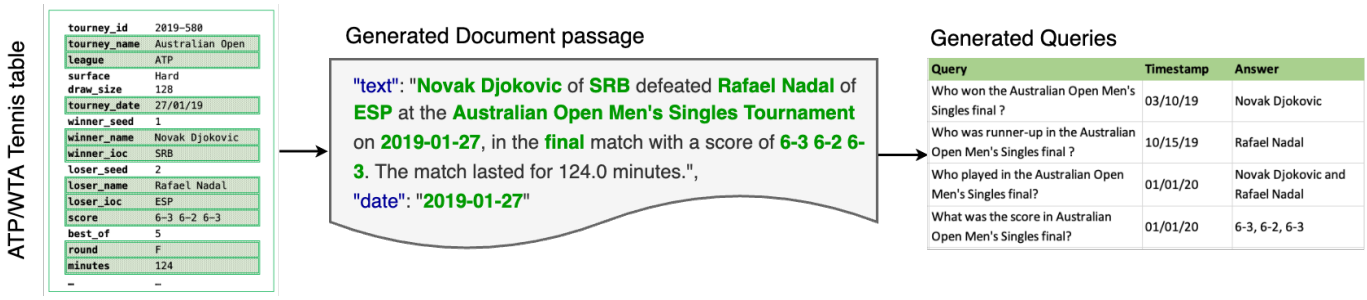

Fig. 2. An example of converting tabular data into a text passage queries we generate based on the passage. The timestamp element enables us to simulate queries that relate to the same event but are posed at different times.

TABLE I
DATASET DETAILS - WE USE TABLEPEDIA GENERATED DATASETS FOR THE INDEX, AND TEMPORAL PROXIMITY QUERY SETS FOR EVALUATION.

Dataset	ATP/WTA Tennis	Oscars Awards
Passages	50479	10267
Time range of questions	1978 - 2019	1922-2019
Few Shot Training Sizes	32, 64, 128	32, 64, 128
Eval Set 1 (TPQ - 2019)	128 questions, asked in 2019	
Eval Set 2 (TPQ - 2020)	128 questions, asked in 2020	

template, covering tournament names, winner names, date of match and final scores. We also extracted the timestamp from the row, in the format, $YYYY - MM - DD$ and attached it to the passage as metadata. The process is illustrated in Figure 2. The key characteristics of both our tennis-based and Oscars-based datasets are shown in Table I.

We create two custom document indices with our TablePedia methodology, to demonstrate tranferability of our methodology across domains, and use them to evaluate TempRALM method on ATLAS and GPT-3.5 with RALM extensions:

- **TablePedia-Tennis** is a TablePedia generated dataset based on the kaggle ATP/WTA table [12], which covers Men's and Women's tennis grand slam tournaments from 1978 to 2019, and contains around 50,000 passages. A sample passage is shown in Figure 2.
- **TablePedia-Oscars** was retrieved from a kaggle-based tabular dataset listing Oscars winners and nominees from 1922-2019 [23]. We used the Tablepedia methodology to generate a document index from it, which consists of roughly 10,000 passages.

We show statistics related to Tablepedia-Tennis and Tablepedia-Oscars in Table I

B. Training and Evaluation Data

As our focus is on temporally aware question-answering, we use the input data format (query, timestamp, answer), where the timestamp indicates the date and time a query is posed, and the answer contains the information valid at that specific time.

For instance, given the query "Who won the Australian Open Women's singles final?" posed on May 14, 2012, the correct response should indicate the player who held the title of Australian Open Women's singles champion as of May 14, 2012. It is crucial to note that the response to the same query changes as soon as a new Australian Open tournament completes. We discuss the details of the training and evaluation sets next.

1) Training Set: We focus on temporally-aware question answering using an input format of (query, timestamp, answer). For each passage, we generate query-answer pairs with a timestamp indicating when the query is posed in order to be able to verify the validity of the answer relative to the timestamp of the question. Our training/fine-tuning datasets are outlined below:

- **Few-shot training set for ATLAS**. We create 3 non-overlapping few-shot training sets of sizes 32, 64 and 128, each represented as a triple in the format (query, timestamp, answer). These are the recommended training set sizes for ATLAS with most use cases requiring at least 32 few-shot examples, and showing good performance at 64 or 128 examples. We provide additional details of this setup in Section V-A.
- **Few-shot prompts for GPT-3.5 RALM**. For our experiment with GPT-3.5 which we augmented with RALM capabilities, we generate 4 triplets in the format (query, passages, correct-passage) for prompting the model to generate an answer. We show examples of these few shot prompts in Table II.

2) Test sets: We created two evaluation test sets, each consisting of 128 ground truth triplets, following the same format and content as the training set. We refer to these tests as Temporal Proximity Query sets (TPQs), labeled TPQ-2019 and TPQ-2020, focus on tournament data from 2019 with queries being posed in 2019 and 2020 respectively. As illustrated in Figure 3, both query sets contain identical questions about matches that occurred in 2019 but have timestamps from two consecutive years, 2019 and 2020, simulating two temporal contexts. For example, given the question "Who won the US Open Women's singles final?", TPQ-2019 timestamps the query to 2019 (e.g., 12-31-2019), while TPQ-2020 timestamps it to 2020 (e.g., 01-01-2020). In both cases, the answer corresponds to the US Open tournament held in September 2019. Correctly answering the question requires the model

TABLE II

FEW-SHOT PROMPTS TO GPT-3.5 RAG: THIS TABLE SHOWS THE PROMPTS (TENNIS USE CASE), WHERE WE ASK A QUESTION AND PROVIDE 3 PASSAGES TO CHOOSE FROM, ALONG WITH THE CORRECT ANSWER.

Role	Content
System	I am a very helpful agent. Given a QUESTION and PASSAGES, I read the PASSAGES carefully and ANSWER the QUESTION. I always provide an ANSWER.
User	PASSAGES: Simona Halep of ROU defeated Angelique Kerber of GER at the Australian Open Women's Singles Tournament on 2018-01-15, in the semi final match with a score of 6-3 4-6 9-7. Victoria Azarenka of BLR defeated Na Li of CHN at the Australian Open Women's Singles Tournament on 2013-01-14, in the final match with a score of 4-6 6-4 6-3. Serena Williams of USA defeated Maria Sharapova of RUS at the Australian Open Women's Singles Tournament on 2015-01-19, in the final match with a score of 6-3 7-6(5). QUESTION: What was the final match score in Australian Open Women's Singles Tournament as of 2013-12-24?
System	4-6 6-4 6-3
User	PASSAGES: Martina Hingis of SUI defeated Anna Kournikova of RUS at the Australian Open Women's Singles Tournament on 1998-01-19, in the third round match with a score of 6-4 4-6 6-4. Serena Williams of USA defeated Na Li of CHN at the Australian Open Women's Singles Tournament on 2010-01-18, in the semi final match with a score of 7-6(4) 7-6(1). Venus Williams of USA defeated Casey Dellacqua of AUS at the Australian Open Women's Singles Tournament on 2010-01-18, in the third round match with a score of 6-1 7-6(4). QUESTION: Who won the Australian Open Women's Singles Tournament as of 1998-10-18?
System	Martina Hingis
User	PASSAGES: Rosana De Los Rios of PAR defeated Anne Gaelle Sidot of FRA at the Roland Garros Women's Singles Tournament on 2002-05-27, in the second round match with a score of 6-3 6-1. Simona Halep of ROU defeated Angelique Kerber of GER at the Roland Garros Women's Singles Tournament on 2018-05-28, in the quarter finals match with a score of 6-7(2) 6-3 6-2. Sloane Stephens of USA defeated Darya Kasatkina of RUS at the Roland Garros Women's Singles Tournament on 2018-05-28, in the quarter finals match with a score of 6-3 6-1. QUESTION: Who was the runner-up in the Roland Garros Women's Singles Tournament as of 2017-12-22?
System	Simona Halep
User	PASSAGES: Andy Murray of GBR defeated Roger Federer of SUI at the Australian Open Men's Singles Tournament on 2013-01-14, in the semi final match with a score of 6-4 6-7(5) 6-3 6-7(2) 6-2. Andre Agassi of USA defeated Rainer Schuettler of GER at the Australian Open Men's Singles Tournament on 2005-01-17, in the second round match with a score of 6-3 6-1 6-0. Rafael Nadal of ESP defeated Roger Federer of SUI at the Australian Open Men's Singles Tournament on 2009-01-19, in the final match with a score of 7-5 3-6 7-6(3) 3-6 6-2. QUESTION: Who played in the Australian Open Men's Singles final as of 2009-12-23?
System	Rafael Nadal and Roger Federer

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. Example of temporal awareness in answering a query based on the timestamp.

to go beyond simple pattern matching (of character strings forming dates) and leverage temporal awareness, enabling it to assess the proximity between dates.

V. EXPERIMENTS

We conduct experiments using two baseline retrieval-augmented approaches - the open-source ATLAS model and the commercial LLM, GPT-3.5, which we augment with RAG. We augment both models with our TempRALM extensions and evaluate their performance relative to their baseline configurations in the context of our TablePedia-generated tennis and Oscars datasets (Section IV) using the TPQ-2019 and TPQ-2020 test sets (Section IV-B2).

A. TempRALM-augmented ATLAS

We use the ATLAS-large model [10] (770M reader / 110M retriever parameters) as our baseline, and compare it against a version augmented with our temporal extensions, which we refer to as TempRALM-ATLAS.

We create three non-overlapping few-shot training sets of sizes 32, 64, and 128, structured in the (query, timestamp, answer) format, for the few-shot training of ATLAS (Section IV-B1). Each query in these sets has a timestamp prior to 2019 to avoid overlap with the test sets. For both datasets, we train ATLAS and TempRALM-ATLAS across all three few-shot training set sizes and evaluate them on the TPQ-2019 and TPQ-2020 test sets using an exact match metric (Figure 1). Each experiment is run five times and the results are averaged. We conducted our experiments using a single NVIDIA A100-

PCIE-40GB node at a university high performance GPU cluster.

We find that in cases where the query and passage years differ (TPQ-2020), TempRALM-ATLAS outperforms ATLAS across the board, including by 49% in the 32-shot, 67% in the 64-shot, and 74% in the 128-shot experiments (Table III). When the query year matches the event year (TPQ-2019 test set), TempRALM-ATLAS performs comparably to ATLAS, as the semantic score dominates the retrieval.

Our results also show that as the number of few-shot training examples increases, TempRALM's impact increases. Consistent with [10], we observe that performance increases more slowly as the number of few-shot examples increases.

To assess whether TempRALM enhances ATLAS performance across different model sizes, we implemented it on both the ATLAS-large model (770M reader/110M retriever) and the ATLAS-base model (220M reader/110M retriever). Our results show that TempRALM improves performance across both model sizes (Figure 4).

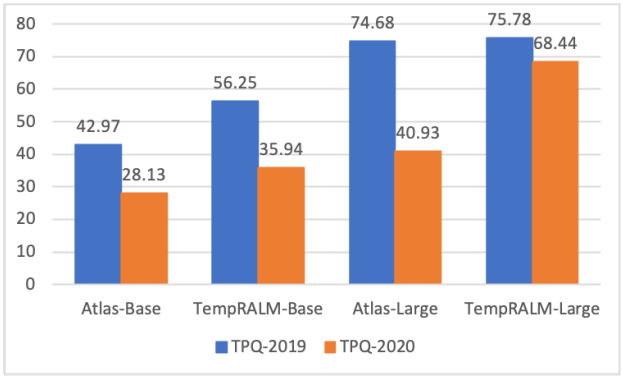

Fig. 4. Comparison of exact match results in a 64-shot setting between Atlas and TempRALM in base vs. large model configurations (tennis use case)

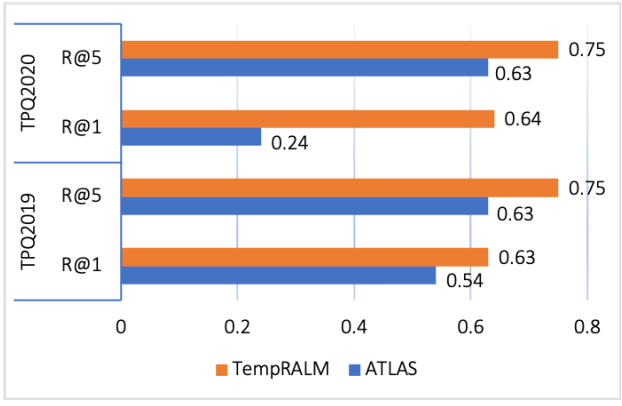

Fig. 5. Comparison of Retriever Recall in a 64-shot training setting (tennis use case).

Finally, in order to better understand the improvement in the (passage) retrieval performance, we look at recall@1 and recall@5 (top 1 and top 5 retrieved passages respectively), and note significant improvements achieved by TempRALM (Fig. 5), demonstrating the effectiveness of the temporal scores at correctly ranking the temporally-relevant passages in the $top\text{-}k$ returned results.

Figure 6 provides an illustrates of how each retrieval approach (ATLAS, TempRALM-ATLAS) generates responses to queries from the TPQ-2020 test set.

The key findings from our experiments are as follows:

- TempRALM-ATLAS consistently performs better than ATLAS across both datasets and few shot set sizes indicating the transferability of our method across different domains.
- TempRALM-ATLAS performs significantly better on the TPQ-2020 test sets by enhancing the retrieval with temporally accurate passages, as well as improving the R@1 and R@5 recall for both TPQ test sets.
- TempRALM shows its effectiveness on both model sizes (base and large) demonstrating the robustness of the methodology.

B. TempRALM-augmented commercial LLM

We enhanced GPT-3.5 [11] with Retrieval-Augmented Generation (RAG) [6] by providing the model with a question and additionally a set of relevant passages to serve as context, thus simulating a RAG scenario. We used the OpenAI embeddings API [24] to encode each query and generate the document index. The semantic similarity between the query and passages is computed as described in equation 1, and used to retrieve the $top\text{-}k$ passages, which are then fed into the GPT model along with the query. We refer to this model as GPT-RAG. Additionally, we incorporated our temporal extensions into GPT-RAG, resulting in the TempRALM-GPT-RAG model. In the following section, we present a comparison of both models (GPT-RAG and TempRALM-GPT-RAG) which enables us to evaluate the impact of TempRALM on GPT-RAG's performance.

We selected four triplets in the format (query, passages, answer) to encompass all question types in our datasets (Section IV-B1). We refer to this few-shot learning with 4 examples as 4-shot learning. Each triplet is a question along with 3 passages, where one passage contains the correct answer to the question. We show how we structure our prompts with an appropriate role in Table II. These triplets are used to assess model performance under both 0-shot and few-shot prompting conditions. We also investigate the effect of varying embedding sizes (large vs. small) on model performance.

Table IV presents our performance comparison between GPT-RAG and TempRALM-GPT-RAG evaluated in the context of the tennis and Oscars datasets. We report the performance metrics for both small (SEM) and large (LEM) embeddings across various experimental configurations in order to understand the impact of embedding size on performance.

We see a notable performance gain for GPT-3.5 when enhanced with TEMPRALM, especially on TPQ-2020, where standard retrieval methods often struggle due to the complexity of temporally-dependent queries. This improvement is especially significant in the Oscars dataset, where TEMPRALM's

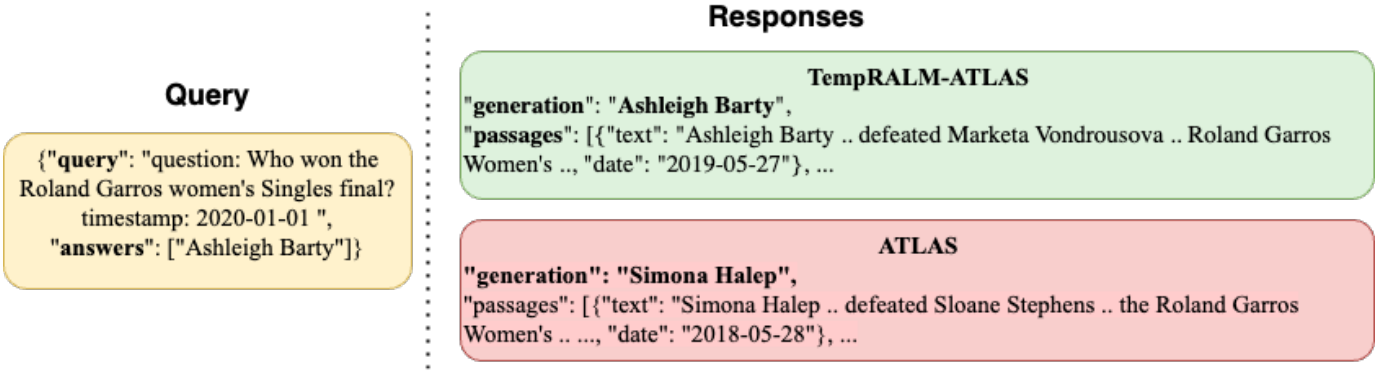

Fig. 6. Examples of responses generated for a TPQ-2020 question by ATLAS, and TempRALM-ATLAS

TABLE III
PERFORMANCE COMPARISON OF THE ATLAS-LARGE MODEL AND TEMPRALM-ATLAS ON TPQ-2019 AND TPQ-2020, FINE-TUNED WITH LIMITED (FEW-SHOT) TRAINING DATA (32, 64, AND 128 TIME-SUFFIXED QA PAIRS).

		Tennis dataset			Oscar dataset		
		32-shot	64-shot	128-shot	32-shot	64-shot	128-shot
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**	–
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 IV
PERFORMANCE COMPARISON OF GPT-3.5 WITH RAG VS. GPT-3.5 WITH TEMPORALLY-AUGMENTED RAG (SEM: SMALL EMBEDDINGS, LEM: LARGE EMBEDDINGS).

	Tennis		Oscars	
	TPQ-2019	TPQ-2020	TPQ-2019	TPQ-2020
GPT-RAG (sem)	34.38	12.5	–	–
TempRALM-GPT-RAG (sem)	**59.38**	**43.75**	–	–
GPT-RAG (lem)	43.8	43.75	65.38	57.69
TempRALM-GPT-RAG (lem)	**75**	**56.25**	**73.07**	**76.92**
GPT-RAG + 4-shot (lem)	90.63	50	73.07	42.03
TempRALM-GPT-RAG + 4-shot (lem)	**93.75**	**65.63**	**84.61**	**76.92**

temporal mechanisms lead to an increase of 34 percentage points for TPQ-2020, compared to a 16-point increase in the tennis dataset. This shows that TEMPRALM's temporal handling capability can effectively adapt to use cases from different domains.

We also notice a difference in results between the tennis and Oscars datasets which suggests that TEMPRALM performs better on data with richer contextual variety, such as movie names and award categories. This variability contrasts with the tennis dataset, where top players often appear as winners across multiple years. The Oscars dataset provides a clearer separation of events by year, allowing TEMPRALM to leverage temporal nuances more effectively.

The key findings from our experiments are as follows:

- TempRALM-GPT-RAG consistently performs better than GPT-RAG across all embedding sizes, across datasets, demonstrating the effectiveness of TempRALM regardless of model size and dataset domain.
- Using large embeddings (LEM) results in better performance than small embeddings (SEM) in both models.
- The 4-shot TempRALM-GPT-RAG+4-shot (lem) model shows significant gains over TempRALM-GPT-RAG demonstrating the effectiveness of providing few-shot examples in the prompts.

VI. CONCLUSIONS

In this study, we introduced and evaluated TempRALM, a temporally-aware augmentation for retrieval-augmented language models (RALMs). 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 improvement without the need for model pre-training, replacing the document index with an updated index, or adding any other computationally intensive elements. In addition, we presented a novel automated method for generating factual documents from tables, which does not require manual validation. We call this method TablePedia and demonstrate its utility across a wide range of domains by generating datasets for two very different use cases: tennis and Oscars awards and using them to evaluate TempRALM.

Our work improves the retrieval of time-sensitive information, which is essential for applications across a broad range of domains, including finance, business intelligence, and healthcare. By integrating temporal awareness into retrieval models, we ensure that the retrieved information is both contextually relevant and timely. Our approach aligns with human-centered AI principles by prioritizing the real-world needs of users, and supporting decision-making processes where accurate and up-to-date information is critical.

VII. Future Work

In the future, we plan to build on this work along a number of dimensions, including implementing and evaluating different learning strategies for optimizing the parameters of our temporal relevance function, exploring the interplay between the retriever and the LLM, and leveraging implicit timestamps embedded within documents. Additionally, we plan to apply our methods to diverse datasets and complex scenarios where temporal awareness can enhance retrieval, such as real-time data, social media posts, news platforms, and contexts involving both external and internal temporal factors. Furthermore, temporally-aware mechanisms have the potential to improve the performance of multimodal systems by enabling the retrieval of images and videos based on the time they were created.

References

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, 2020.

[2] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, E. Chu, J. H. Clark, L. E. Shafey, Y. Huang, K. Meier-Hellstern, G. Mishra, E. Moreira, M. Omernick, K. Robinson, S. Ruder, Y. Tay, K. Xiao, Y. Xu, Y. Zhang, G. H. Abrego, J. Ahn, J. Austin, P. Barham, J. Botha, J. Bradbury, S. Brahma, K. Brooks, M. Catasta, Y. Cheng, C. Cherry, C. A. Choquette-Choo, A. Chowdhery, C. Crepy, S. Dave, M. Dehghani, S. Dev, J. Devlin, M. Díaz, N. Du, E. Dyer, V. Feinberg, F. Feng, V. Fienber, M. Freitag, X. Garcia, S. Gehrmann, L. Gonzalez, G. Gur-Ari, S. Hand, H. Hashemi, L. Hou, J. Howland, A. Hu, J. Hui, J. Hurwitz, M. Isard, A. Ittycheriah, M. Jagielski, W. Jia, K. Kenealy, M. Krikun, S. Kudugunta, C. Lan, K. Lee, B. Lee, E. Li, M. Li, W. Li, Y. Li, J. Li, H. Lim, H. Lin, Z. Liu, F. Liu, M. Maggioni, A. Mahendru, J. Maynez, V. Misra, M. Moussalem, Z. Nado, J. Nham, E. Ni, A. Nystrom, A. Parrish, M. Pellat, M. Polacek, A. Polozov, R. Pope, S. Qiao, E. Reif, B. Richter, P. Riley, A. C. Ros, A. Roy, B. Saeta, R. Samuel, R. Shelby, A. Slone, D. Smilkov, D. R. So, D. Sohn, S. Tokumine, D. Valter, V. Vasudevan, K. Vodrahalli, X. Wang, P. Wang, Z. Wang, T. Wang, J. Wieting, Y. Wu, K. Xu, Y. Xu, L. Xue, P. Yin, J. Yu, Q. Zhang, S. Zheng, C. Zheng, W. Zhou, D. Zhou, S. Petrov, and Y. Wu, "Palm 2 technical report," 2023. [Online]. Available: https://arxiv.org/abs/2305.10403

[3] OpenAI, "Chatgpt." [Online]. Available: https://openai.com/index/chatgpt/

[4] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu, Z. Wu, L. Zhao, D. Zhu, X. Li, N. Qiang, D. Shen, T. Liu, and B. Ge, "Summary of chatgpt-related research and perspective towards the future of large language models," *Meta-Radiology*, vol. 1, no. 2, p. 100017, sep 2023. [Online]. Available: http://dx.doi.org/10.1016/j.metrad.2023.100017

[5] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, and S. Riedel, "Language models as knowledge bases?" 2019. [Online]. Available: https://arxiv.org/abs/1909.01066

[6] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "REALM: retrieval-augmented language model pre-training," *CoRR*, vol. abs/2002.08909, 2020. [Online]. Available: https://arxiv.org/abs/2002.08909

[7] K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston, "Retrieval augmentation reduces hallucination in conversation," *CoRR*, vol. abs/2104.07567, 2021. [Online]. Available: https://arxiv.org/abs/2104.07567

[8] B. Dhingra, J. R. Cole, J. M. Eisenschlos, D. Gillick, J. Eisenstein, and W. W. Cohen, "Time-aware language models as temporal knowledge bases," *CoRR*, vol. abs/2106.15110, 2021.

[9] O. Sharir, B. Peleg, and Y. Shoham, "The cost of training NLP models: A concise overview," *CoRR*, vol. abs/2004.08900, 2020. [Online]. Available: https://arxiv.org/abs/2004.08900

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

[11] OpenAI, "Gpt 3.5." [Online]. Available: https://platform.openai.com/docs/models/o1#gpt-3-5-turbo

[12] kaggle taylorbrownlow, "atpwta-tennis-data." [Online]. Available: https://www.kaggle.com/datasets/taylorbrownlow/atpwta-tennis-data

[13] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave, "Towards unsupervised dense information retrieval with contrastive learning," 2021. [Online]. Available: https://arxiv.org/abs/2112.09118

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[15] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2023. [Online]. Available: https://arxiv.org/abs/1910.10683

[16] M. de Jong, Y. Zemlyanskiy, J. Ainslie, N. FitzGerald, S. Sanghai, F. Sha, and W. Cohen, "Fido: Fusion-in-decoder optimized for stronger performance and faster inference," 2023. [Online]. Available: https://arxiv.org/abs/2212.08153

[17] W. X. Zhao, J. Liu, R. Ren, and J.-R. Wen, "Dense text retrieval based on pretrained language models: A survey," 2022.

[18] M. J. Q. Zhang and E. Choi, "Situatedqa: Incorporating extra-linguistic contexts into qa," 2021. [Online]. Available: https://arxiv.org/abs/2109.06157

[19] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. tau Yih, "Dense passage retrieval for open-domain question answering," 2020. [Online]. Available: https://arxiv.org/abs/2004.04906

[20] J. Kasai, K. Sakaguchi, Y. Takahashi, R. L. Bras, A. Asai, X. Yu, D. Radev, N. A. Smith, Y. Choi, and K. Inui, "Realtime qa: What's the answer right now?" 2024. [Online]. Available: https://arxiv.org/abs/2207.13332

[21] J. Jang, S. Ye, C. Lee, S. Yang, J. Shin, J. Han, G. Kim, and M. Seo, "Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models," 2023.

[22] wiki/Standard_score, "Standard_score." [Online]. Available: https://en.wikipedia.org/wiki/Standard_score

[23] kaggle unanimad, "the-oscar-award." [Online]. Available: https://www.kaggle.com/datasets/unanimad/the-oscar-award/data

[24] OpenAI, "Embeddings guide," 2024, accessed: 2024-11-03. [Online]. Available: https://platform.openai.com/docs/guides/embeddings