

Evaluating List Construction and Temporal Understanding capabilities of Large Language Models

Dumitru, Alexandru; Venkatesh, V.; Jatowt, Adam; Anand, Avishek

DOI

[10.1145/3731120.3744606](https://doi.org/10.1145/3731120.3744606)

Licence

CC BY

Publication date

2025

Document Version

Final published version

Published in

ICTIR 2025 - Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval

Citation (APA)

Dumitru, A., Venkatesh, V., Jatowt, A., & Anand, A. (2025). Evaluating List Construction and Temporal Understanding capabilities of Large Language Models. In *ICTIR 2025 - Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval* (pp. 369-379). (ICTIR 2025 - Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval). Association for Computing Machinery (ACM). <https://doi.org/10.1145/3731120.3744606>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.



Evaluating List Construction and Temporal Understanding capabilities of Large Language Models

Alexandru Dumitru*

alexandru.dumitru@prosus.com

Prosus

Delft, Netherlands

Adam Jatowt

adam.jatowt@uibk.ac.at

University of Innsbruck

Innsbruck, Austria

Venktesh V*

v.viswanathan-1@tudelft.nl

Delft University of Technology

Delft, Netherlands

Avishek Anand

avishek.anand@tudelft.nl

Delft University of Technology

Delft, Netherlands

Abstract

Large Language Models (LLMs) have demonstrated immense advances in a wide range of natural language tasks. However, these models are susceptible to hallucinations and errors on particularly temporal understanding tasks involving multiple entities in answers. In such tasks, they fail to associate entities with accurate time intervals, generate a complete list of entities in answers or reason about events associated with specific temporal bounds. Existing works do not extensively evaluate the abilities of the model to perform implicit and explicit temporal understanding in a list answer construction setup. To bridge this gap, we propose the *Time referenced List based Question Answering* or TLQA benchmark that requires structured answers in list format aligned with corresponding time periods. Our TLQA benchmark, requires both list construction and temporal understanding simultaneously, which to the best of our knowledge has not been explored in prior benchmarks. We investigate the temporal understanding and list construction capabilities of state-of-the-art generative models on TLQA in closed-book and open-domain settings. Our findings reveal significant shortcomings in current models, particularly their inability to provide complete answers and temporally align facts in a closed-book setup and the need to improve retrieval in open-domain setup, providing clear future directions for research on TLQA. The benchmark and code at <https://github.com/elixir-research-group/TLQA>.

CCS Concepts

• Information systems → Question answering.

Keywords

Temporal Question Answering, Retrieval, Temporal Understanding

ACM Reference Format:

Alexandru Dumitru*, Venkatesh V*, Adam Jatowt, and Avishek Anand. 2025. Evaluating List Construction and Temporal Understanding capabilities of Large Language Models. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval*



This work is licensed under a Creative Commons Attribution 4.0 International License. *ICTIR '25, Padua, Italy*

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1861-8/2025/07

<https://doi.org/10.1145/3731120.3744606>

(*ICTIR*) (*ICTIR '25*), July 18, 2025, Padua, Italy. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3731120.3744606>

1 Introduction

Large Language Models have made tremendous advances in wide range of Natural language processing (NLP) tasks [2, 27, 32]. Their ability to reason and answer questions with factual information has been studied widely through well known Question Answering (QA) benchmarks [12, 28, 34, 38]. However, such models are still prone to errors and hallucination [15] where they generate plausible sounding answers not grounded on facts in the real world. These

Example: TLQA and extensions

TLQA: List all political positions Joe Biden held to this day.

[Answer]:

☑ President of the United States (2021-2024)

☑ Vice President of the United States (2009-2017)

☑ United States Senator from Delaware (1973-2009)

TLQA-TS question: What political positions were held by Biden between 2009-2020?

[Answer]: ☑ Vice President of the United States (2009-2017)

TLQA-TM question: What political positions were held by Biden after serving as Senator?

[Answer]: ☑ Vice President of the United States (2009-2017) ☑ President of the United States (2021-2024)

Figure 1: An example for Time referenced List based QA

problems are further exacerbated for questions where the model has to generate a structured list of answers (**ListQA**). In such ListQA tasks, the model has to ensure coverage of all relevant entities in the answer (list construction). The ListQA task may include inquiries referring to side effects of certain medication, historical events, priority lists or rankings. Over **10% of Bing's web queries** [4] and healthcare queries [42] **21.9%** of the queries necessitate a structured list as response. However, there have been only few works on queries that require structured, listwise answers [30].

*These authors contributed equally to this work

Additionally, factual information of entities evolves over time and requires reasoning about the scope of knowledge in different time periods. Hence, the LLM must possess **temporal understanding** capabilities such as associating the entities in the list answer with accurate time intervals and also reason about temporal scope of the entity in the question. This may also entail performing implicit temporal understanding for questions like “What positions were held by Joe Biden after his tenure as senator?”. Here the LLM has to infer the time period of position as senator and perform temporal arithmetic to arrive at the time period after the official left this position to arrive at the correct answer. While existing works have focused on questions with temporal markers, they were primarily based on Knowledge Graphs (KG) [7, 16, 17, 31, 37] and not text-based, which limits the understanding of temporal evolution and transitions in text. The emphasis on KG based temporal facts has been found to be limiting research as it ignores implicit temporal structure or temporal reasoning tasks beyond KGs. Further, prior works have observed that **heuristics or shortcuts** can often answer these questions without necessitating genuine temporal reasoning [8]. While few works focus on temporal understanding over natural text [10, 29], these works do not address questions that require multiple answers (list based), which is common in real-world scenarios involving time-period-specific queries.

To bridge this gap, we study time-referenced List based QA, by curating a benchmark (TLQA) along with **large evidence collection** of Wikipedia articles and corresponding info boxes. The questions in TLQA require a list of answers each associated with a time period. This reflects real-world queries where users are interested in querying historical events or news [41]. These are common queries issued to search engines which are time-sensitive or involve temporal specifiers [6, 16] such as “What teams did LeBron James play for between 2007-2009?”. These queries are complex to resolve as they need to not only maintain completeness of the answer list (**list construction** ability) but also get accurate bounds (start and end) of the related time periods and ability to associate entities with the accurate time periods (**temporal understanding** ability). Some queries in our benchmark, such as the TLQA-TM question in Figure 1 also require implicit temporal understanding capability to infer time period related to the event specified in the question. Examples of diverse question types in our benchmark are presented in Figure 1. Unlike typical QA scenarios such as those in SQuAD [33] where the system extracts a single answer from a single body of text, TLQA systems must find multiple answers from a single text or multiple sources [43] and align them with appropriate time periods while ensuring the bounds for time intervals are accurate. We would also like to note that exact sorting of time periods is beyond scope, though our benchmark can be used for the same.

We evaluate a range of generative Large Language Models (LLMs) in few-shot closed book setup and in open-domain setup to analyze their **temporal understanding** and their **list construction** abilities. To the best of our knowledge, prior works have not tackled the temporal understanding and list structured output abilities simultaneously, critical for real-world applications.

Real-world applications: Benchmarking LLMs on TLQA can help improve their temporal understanding and list construction capabilities, which is of immense use in healthcare, historical research and education and journalism. For instance, in historical

research/education it helps scholars access precise historical information regarding different events across time. In healthcare, it could be used to access historical medical records.

Research Questions:

RQ1: How well do Large Language Models (LLMs) perform on temporal understanding and list construction based questions in TLQA ?

RQ2: Does retrieval augmentation from external knowledge sources help reduce hallucination in LLMs for generating complete lists in answers with precise temporal bounds?

RQ3: When provided with golden evidence in retrieval augmented setup, what are the effects of various distractors (as evidence) on model performance?

2 Related Work

While several ListQA benchmarks have been proposed, they primarily focus on answering factoid or ambiguous questions with multiple answers [1, 25, 46]. They do not focus on the evolution of such factual answers with shift in temporal information. Temporal information plays an important role for real-world tasks such as Information Retrieval and Question Answering [6, 45]. Information evolves over time [11, 14], and it is critical to provide temporally recent and relevant information to users. Hence, temporal information retrieval [3, 21] and temporal QA [37, 39] has been of immense interest recently. These approaches commonly use temporal signals in text to ascertain temporal aspects of query intents [18], perform query or document matching [13], aiding in search of web archives [19]. A large body of studies have been dedicated to study evolution of facts through temporal knowledge graphs (TKG) [5, 16, 17, 44] and evaluate QA over such graphs [35]. However, [8] observed that such datasets consists of primarily pseudo-temporal questions where QA tasks could be solved without enforcing temporal constraints. To tackle this, MultiTQ [7] mandates temporal constraints for QA.

However, answering questions over temporal knowledge graphs are limited to the facts contained in the constructed knowledge graph. To study temporal evolution of facts in natural text, several datasets [6, 22] focus on time aware QA with textual evidence. More recently, **TempLAMA** [10] introduced cloze type questions to test the temporal awareness of pre-trained language models. However, it has several limitations, such as answers with missing start date and 7.9% of the questions have subjects with incomplete names, which can be problematic for disambiguation such as ‘Cristiano Ronaldo’ and ‘Ronaldo.’. Additionally, the evaluation metrics employed in TempLAMA such as maximum token-level F1 score does not account for answer completeness and temporal bounds.

3 Benchmark Creation

3.1 Time Referenced List based QA

Definition 3.1. A time-referenced list based question Q is a query that, given a temporal context t , requests a comprehensive list of entities or facts e constituting together a correct answer to Q over the time period t . The answer A to Q is a set of pairs (e, τ) where e is an entity or fact, and τ is the time interval when e is relevant to Q .

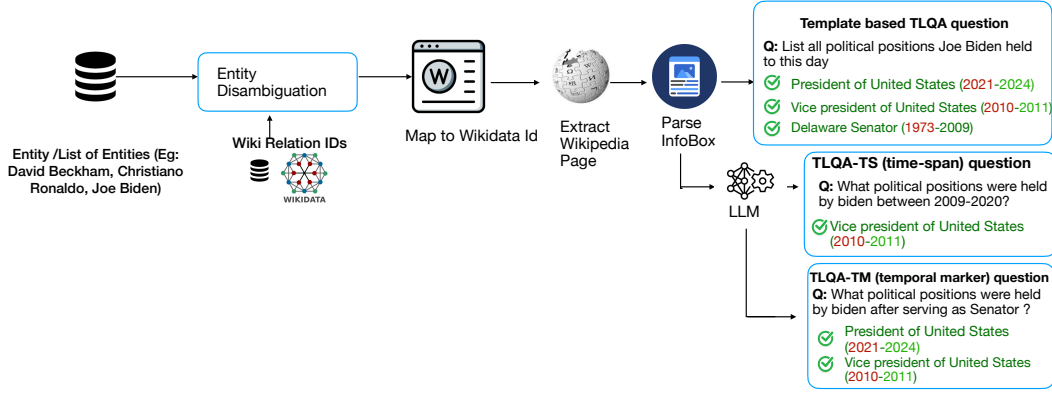


Figure 2: TLQA data collection pipeline

3.2 TLQA generation

We propose an automated solution to generate TLQA. Our focus is generating questions from entities related to Wikidata relations namely P54 (member of a sports team) and P39 (position held). The main intuition behind this decision is that only these topics, the entities are naturally associated with multiple organizations/positions over different time periods, making it a natural choice for TLQA. Our data-curation pipeline consists of multiple stages such as entity/subject extraction, Wikipedia Mapping, Infobox Extraction and question generation with templates shown in Figure 2.

3.2.1 Entity seed set. Our pipeline first starts with a list of entities/subjects which would form the core of the questions. We extract these entities from TempLAMA to act as **seed set** though the pipeline would work with any set of entities. Hence, we would like to note that our pipeline is generic **without dependency** on TempLAMA and requires only a list of entities or subjects with relation pairs. For each query in TempLAMA, the subject of each query is extracted by using the Wikidata relation type as a semantic marker. For example, for relation P54, each query is of the form subject plays for _X_, and can be split based on the words plays for, to extract the subject.

3.2.2 Entity Disambiguation and mapping to Wikidata ID. For each subject/entity, the goal is to map to Wikipedia page to extract relevant information and generate questions. Using only the subject's name may be insufficient, as multiple named entities may share the same name. Therefore, to resolve each subject to its correct Wikidata page, we collect Wikidata ID which provides a one-to-one relationship between entities and Wikipedia articles.

To accurately find the Wikidata ID for each subject, we utilize the entity, the Wikidata relation type (e.g., member of sports teams), and the aggregated set of entities related to the subject retrieved using Wikidata relation type, with each related entity having a unique Wikidata identifier. For example, related entities in the context of a sports celebrity could be some of the teams he played for which are objects of the relation type "member of sports teams". We then search for entities with the subject's name on Wikidata, limiting the search to the top 50 results to manage the scope. Next, we filter these results to retain only those entities that have the specified relation type (e.g., P54 for sports teams). Based on the

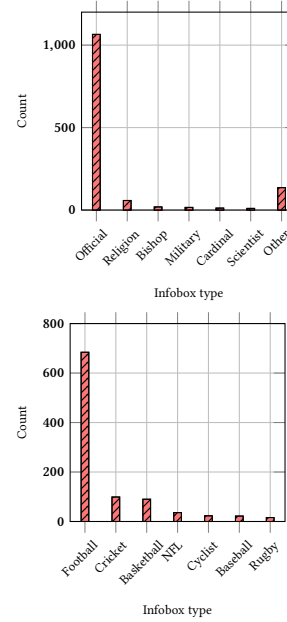


Figure 3: Infobox Type Distribution for P39 and P54

overlap between object IDs in relations for the above candidates and IDs of related entities extracted earlier for the subject, the Wikidata ID is assigned.

After assigning Wikidata IDs, we perform a verification step that checks the label and aliases for each assigned Wikidata ID and compare them with the normalized subject name. If a match is not found among the names, the entity is ignored. This process resulted in the reduction of the initial 5825 entities to 5711 entities.

3.2.3 Infoboxes. After obtaining the Wikidata IDs, we fetch the corresponding infobox for each entity from corresponding Wikipedia pages by parsing the markup. An infobox is a structured table in a Wikipedia article that presents key information about the subject in a standardized format. In Wikipedia's Wikitext, the markup language used by Wikipedia, infoboxes follow specific templates designed for different types of subjects. As of today, there are more

than 1000 types of infoboxes, which can be found on Wikipedia¹. These structures often contain temporal information, which can be collected to form questions that require temporal understanding.

3.2.4 Infobox Distribution over Wikidata relations. We analyze and plot the distribution of infobox types. Infobox types that appear fewer than 10 times are grouped under 'Other'. The plots for some relations are available in Figure 3. We observe that, for the relation P39, **infobox officeholder (official)** represent the most common infobox type, whereas for P54, the most frequent one is **football biography (football)**.

Table 1: Mapping from subject to query based on Wikidata relation type and infobox type

Infobox Type	Temporal Markers	Answer Fields
Football	youth years, years	youth clubs, clubs
Biography	national years	national team
Officeholder	term start term end subterm	office, suboffice jr/sr, state senate state assembly
Cricketer	year, international span	club, country

3.2.5 TLQA from infoboxes. To form TLQA, we construct a question that explicitly requests a list of entities or facts with corresponding time periods associated with the entities. The question templates for infobox types can be seen in Table 2.

Table 2: Fields for temporal information from Infoboxes

Relation ID	Infobox Type	Query
P54	Football, Cricketer	List all teams <subject> played for to this day.
P39	Officeholder	List all political positions <subject> held to this day.

To create our answer set, given a generated query, we parse the infobox corresponding to each subject and query to extract the relevant information. We utilize the temporal markers mentioned in Table 1 to compile comprehensive answers. For sports-related queries, we extract the teams and the corresponding years the subject played for each team. For political-related queries, we extract the political positions held and the associated time periods. If the end year for a position or team association is not specified, we interpret this as an indication that the subject currently holds the position or remains with the team, aligning with the convention used on Wikipedia.

The process of generating questions and answers is applied to the extracted TempLAMA subjects marked with the specified infobox types. This results in a collection of **1655** questions, with an average number of answers of **8.641**. We perform a stratified train-test split based on the question's topic (either political or sports-related). The dataset statistics are shown in Table 3.

¹https://en.wikipedia.org/wiki/Wikipedia:List_of_infoboxes

Table 3: Dataset statistics along with categories.

Category	Split	# of entries	Mean # of Answers
Political	Training	630	5.473
	Test	251	5.438
	Total	881	5.463
Sports	Training	528	11.775
	Test	246	11.915
	Total	774	11.820

Table 4: Manual eval. GAC: Ground Truth Answer Completeness, QU : Question usefulness. We use the Likert scale (1-5) and Cohen's Kappa (κ) for inter-annotator agreement (in brackets).

GAC (κ)	QU (κ)
4.75±0.50 (0.67)	4.98±0.05 (0.74)

Prompt: Generate Temporal-Span based QA TLQA-TS

► **SYSTEM MESSAGE:**

- You will receive an original question along with its correct answer. The question asks for a list of entities associated with a person, including temporal information.
- Your task is to **generate a new question-answer pair** by rephrasing the original question to include a specific time interval condition, such as "between 2000 and 2010", "before 2000", or "after 2010".
- Select a time interval that includes as many entities as possible from the original answer (i.e., maximize overlap with the time-lines). The answer should be a subset of the original answer, including only those entities whose time spans overlap with the selected interval.
- The new question should require temporal reasoning to answer due to the added time constraints.
- The answer should maintain the same format as the original answer: a list of entities with years denoting the time span.

Output Format:

Provide the response in JSON format, adhering to the schema below:

```
{ "question": "<new question>",
  "answers": { "<entity_1>": "<years>",
               "<entity_2>": "<years>", ...}}
```

Figure 4: Prompt used to generate a new timeline dataset.

3.2.6 Extending TLQA. Since the questions in the benchmark are created based on templates and centered on entities, they may have limited temporal variability and may also not test for implicit temporal understanding capabilities. To address this, we extend TLQA with two additional evaluation subsets (test sets) by generating variations of original questions in TLQA test set. These subsets namely TLQA-TS where TS denotes Time Span and TLQA-TM

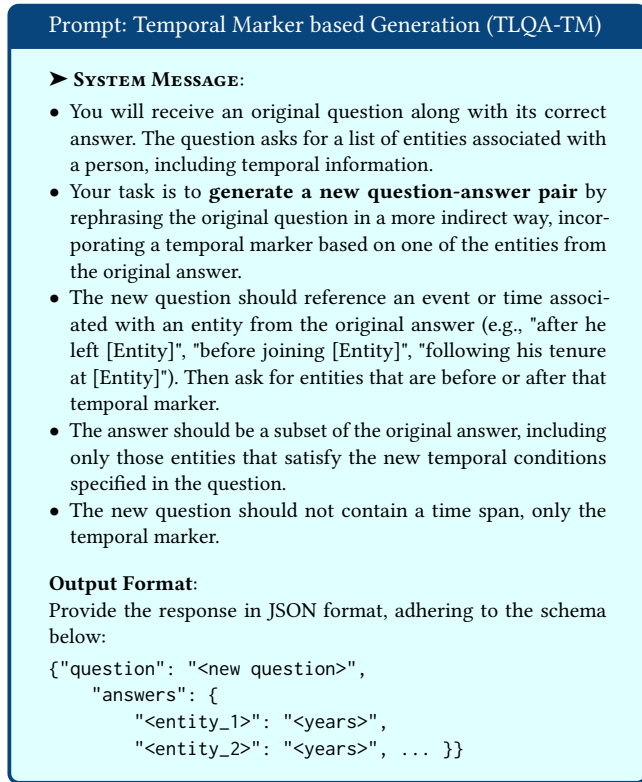


Figure 5: Prompt to generate indirect temporal Q-A pairs.

which denotes Temporal Markers test the ability of the model to reason and provide list-structured answers for different time slices and ability of the model to decipher implicit temporal references and perform temporal understanding respectively. We employ the powerful LLMs GPT4o and using the prompts in Figure 5 (for TM) and Figure 4 (for TS) we generate the new sets of questions. We perform data filtering using the prompt shown in Figure 6, with a powerful GPT4o model as it shown to correlate well with human judgments [40] to validate the correctness of the newly generated questions and answers.

Manual Evaluation: We also perform manual evaluation on 100 questions sampled in a stratified manner by asking annotators to rate the question quality/usefulness and ground truth answer completeness (guidelines in repository) of the generated questions using Likert scale (1-5). The results are shown in Table 4 along with agreement between annotators. The analysis helps validate the quality of our data curation pipeline. This evaluation helps us further filter down to questions that are of high quality. This results in evaluation sets with TLQA-TS comprising 423 and TLQA-TM comprising 460 questions in addition to the 1655 questions.

3.3 Corpus Creation for Open-Domain Setup

We release a corpus which is the latest Wikipedia dump (April 2024)² to evaluate models on TLQA in an open-domain setup. We employ dumpster-dip [23] to parse the Wikipedia dump and extract

²<https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2>

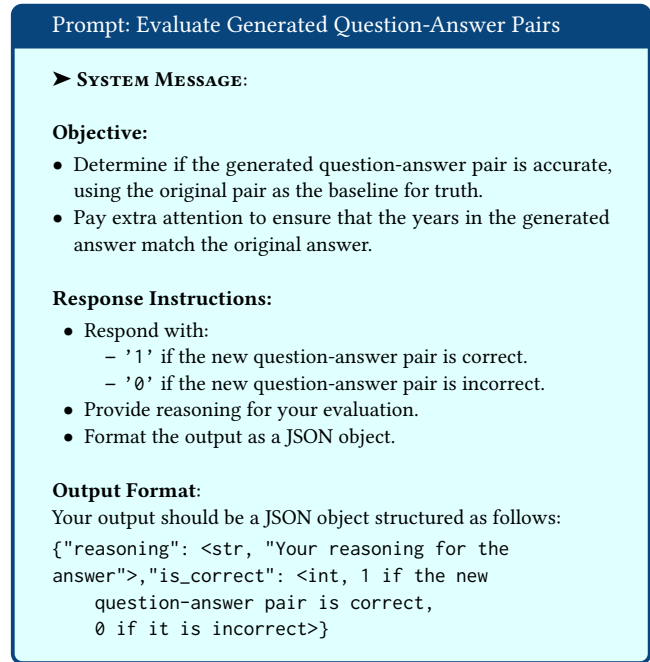


Figure 6: Prompt used for ensuring the quality of automatically generated QA pairs.

semi-structured data such as infoboxes from the articles. After filtering out the articles without an infobox, we get a collection of approximately **4.5 million** articles. For each article, we save the title, infobox, and summary ('lead'³ section of each article).

4 Experimental Setup

We evaluate several LLMs on TLQA (Mistral v0.2, Llama 3.1 8b and GPT-4o-mini). Llama 3.1 underperforms when compared to Mistral v0.2 and hence is not attached in this paper owing to space constraints. However, Llama 3.1 8b results can be found in the repo. We would like to highlight that an *exhaustive evaluation* of all LLMs in the current landscape is not feasible.

4.1 LLM Evaluation Setups

We carry LLM evaluation in (1) **Gold Evidence** setup where, the ground truth infobox is employed to get an upper bound for their performance, (2) **Closed book** setup without external knowledge. (3) **Open-Domain setup** where, we retrieve the relevant evidence (infobox) from the Wikipedia collection (Section 3.3). We consider two scenarios, one where the documents indexed contain *title, infobox, summaries* (title.infobox-summary (T-I-S) setup) and another where the documents contain only title and summaries (title-summary (T-S) setup). In the T-S setup, a mapping between Wikipedia titles and infoboxes is created once and used for lookup during retrieval, to fetch the corresponding infoboxes for retrieved titles.

³https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Lead_section

Table 5: Results of Mistral v0.2 and gpt-4o-mini models on TLQA. The first three metrics evaluate list completeness, the last two assess the temporal correctness (Temporal Overlap (TO) and Temporal Jaccard (TJ))

Model	Precision	Recall	F1	TO	TJ
Closed Book					
Mistral v0.2 7b					
FEW-SHOT	0.558	0.257	0.330	0.384	0.317
KNN FEW-SHOT	0.532	0.310	0.366	0.435	0.360
FEW-SHOT-COT	0.529	0.276	0.338	0.412	0.330
Auto COT	0.523	0.311	0.365	0.346	0.282
GPT 4o-mini					
FEW-SHOT	0.625	0.451	0.501	0.595	0.533
KNN FEW-SHOT	0.620	0.487	0.525	0.610	0.550
FEW-SHOT-COT	0.645	0.459	0.513	0.652	0.578
Auto COT	0.524	0.384	0.420	0.546	0.481
Open Domain					
Mistral v0.2 7b					
BM-25	0.746	0.547	0.607	0.623	0.581
all-mini-lm-v2	0.572	0.380	0.430	0.414	0.366
mutli-qa-mpnet	0.552	0.380	0.441	0.436	0.375
GPT-4o-mini					
BM-25	0.735	0.666	0.685	0.751	0.729
all-mini-lm-v2	0.621	0.528	0.551	0.627	0.578
multi-qa-mpnet	0.581	0.490	0.510	0.593	0.535
Golden Evidence					
Mistral v0.2 7b					
FEW-SHOT	0.918	0.770	0.818	0.747	0.715
KNN FEW-SHOT	0.941	0.850	0.882	0.857	0.820
GPT-4o-mini					
FEW-SHOT	0.916	0.917	0.911	0.950	0.937
KNN FEW-SHOT	0.937	0.950	0.934	0.960	0.949

Hyperparameters: For all experiments, we use a temperature of 0.3 decided based on tuning on validation set.

Evaluation metrics:

Retrieval: Mean Reciprocal Rank (MRR) and Recall@k.

List Construction Metrics: To evaluate list construction performance of LLM outputs, we apply the standard definitions of *Precision*, *Recall*, and F_1 by matching entities in the generated list of answers and the ground truth. To match entities, we follow a three stage pipeline, namely *subsequence* matching and also semantic matching by employing BERT based tokenwise similarity.

Temporal Metrics: Formally, given the set of years covered by the expected date ranges Y_{expected} and the set of years covered by the generated date ranges $Y_{\text{generated}}$, we define Temporal Overlap as:

$$\text{Temporal Overlap} = \frac{|Y_{\text{expected}} \cap Y_{\text{generated}}|}{|Y_{\text{expected}}|}$$

The Temporal Jaccard measures the similarity between the generated and expected date ranges. Unlike Temporal Overlap, Temporal Jaccard penalizes the model for *hallucinating* years outside the expected range. Formally, we define it as: $\text{Temporal Jaccard} = \frac{|Y_{\text{expected}} \cap Y_{\text{generated}}|}{|Y_{\text{expected}} \cup Y_{\text{generated}}|}$. We first parse and normalize the date ranges from both the generated and expected answers. For each matched entity, we represent the date ranges as set of years, which we further use to calculate the Temporal Overlap Score and Temporal Jaccard Similarity. We then average these scores across all matched entities.

5 Results

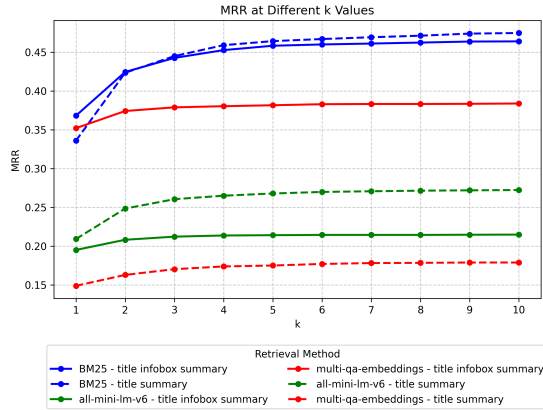
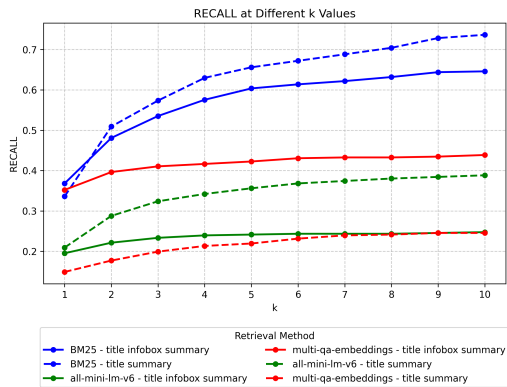
5.1 Performance of LLMs on TLQA

To answer **RQ 1**, we study the performance of Mistral v0.2 and GPT-4o-mini in a closed book setting on TLQA as shown in Table 5 and on TLQA-TS, TLQA-TM in Table 7. From results, we observe that all models have lower listwise recall scores in a closed book setting but have relatively higher precision. We analyzed the answers and observed that this is primarily because the models generate correct answers but incomplete answer lists. We posit that this is primarily due to limitations of model parametric knowledge and additionally instruction tuned models are more precise but however have lower recall, as evidenced by the study [24]. While [24] focuses on free-form answers, precision and recall for our work is measured for lists, providing new insights for list construction abilities of LLMs. Nevertheless, models fine-tuned to follow human instructions tend to lose completeness in their answers.

With respect to **temporal performance**, both models generally perform suboptimally on TLQA and TM, TS subsets. It can be observed that the Time Overlap scores are consistently higher than the Time Jaccard scores, indicating that while the models are not very capable of identifying significant overlapping time periods, they struggle even more with accurately bounding the time-intervals (start and end periods). This is also evident in the results obtained on TLQA-TS subset that requires temporal bounding abilities. This discrepancy likely arises because the models can detect relevant

Table 6: Performance on TLQA for different Retrieval approaches at $k = 1, 3$, and 10 on mistral v0.2.

Retrieval Setup	Corpus	F1@k			Temporal overlap@k			Temporal Jaccard@k		
		k=1	k=3	k=10	k=1	k=3	k=10	k=1	k=3	k=10
all-mini-lm-v2	T-S	0.367	0.430	0.417	0.372	0.414	0.400	0.323	0.366	0.351
	T-I-S	0.394	0.398	0.368	0.410	0.398	0.340	0.360	0.341	0.293
BM25	T-S	0.502	0.594	0.618	0.503	0.594	0.627	0.458	0.556	0.584
	T-I-S	0.536	0.607	0.624	0.558	0.623	0.651	0.513	0.581	0.607
multi-qa-mpnet	T-S	0.353	0.379	0.376	0.373	0.375	0.354	0.310	0.318	0.300
	T-I-S	0.441	0.426	0.400	0.436	0.407	0.388	0.375	0.352	0.326

**Figure 7: Performance of different retrieval settings over the two Wikipedia corpora, T-S and T-I-S using MRR metric. Dashed line is T-S, whereas complete line represents T-I-S.****Figure 8: Comparison of retrieval models (Recall)**

time frames but have difficulty in precisely delineating the start and end years, resulting in either overshooting (example: Table 8) or undershooting the correct periods. We also observe that temporal performance on TLQA-TM is the lowest, as the model is unable to detect the implicit temporal reference made by the marker in the question due to parametric memory limitations (example: Table 8). Among different prompting methods, KNN FEW-SHOT works best.

5.2 Open-Domain Results

Golden Evidence Setup Results: To ascertain the upper bound of performance on TLQA and other subsets, we carry out experiments using LLMs in golden evidence setting where the models are provided with the correct evidence. We observe that this leads to higher performance (Table 5, Table 7) as measured by list construction metrics ($F1=0.934$) and temporal metrics (0.960). Hence, better tabular retrieval will lead to high LLM reasoning performance.

Retrieval Performance: To address RQ 2, we study the impact of augmenting the query with additional evidence to determine if this improves the model’s performance on TLQA. We first evaluate the performance of the different retrieval models in open-domain setup on the two Wikipedia corpora (T-S and T-I-S). The performance of the different retrieval systems (MRR) is shown in Figure 7 and comparison using Recall@k is shown in Figure 8

From the plot, we observe that sparse retrieval approaches like BM25 outperform dense retrieval approaches on both T-S and T-I-S corpora settings. We posit that this is due to the nature of entity centric queries. Sciavolino et al. [36] demonstrate that dense retrieval systems perform considerably worse than BM25 in simple entity-centric questions due to popularity bias.

Retrieval Augmented Answer Generation: We also observe the impact of retrieved documents in an open-domain setting on downstream LLM reasoning, as shown in Table 5 for TLQA. We observe that augmenting LLMs with retrieved documents in general leads to performance gains in list construction and temporal metrics. We observe that retrieval performance directly translates to answer generation performance [26], as BM25 retrieved documents lead to significantly higher gains compared to documents from dense retrieval. For instance, BM25 documents coupled with Mistral v0.2 outperforms all-mini-lm retrieved documents by **45.11%** as measured by $F1$. The results in Table 5 for open-domain setting are carried out with top-3 retrieved documents. We further vary $k=1,3,10$ (number of retrieved documents) and study their impact on downstream LLM based answer generation as shown in Table 6. We observe that LLM reasoning using documents from sparse and dense retrieval approaches, results in slight gains and saturated performance respectively beyond $k=3$. Further, we observe that LLMs are able to decipher implicit temporal references in TLQA-TM when provided with the relevant infobox by mapping the marker to time interval information. However, analysis on TLQA-TS reveals that the temporal bound error persists.

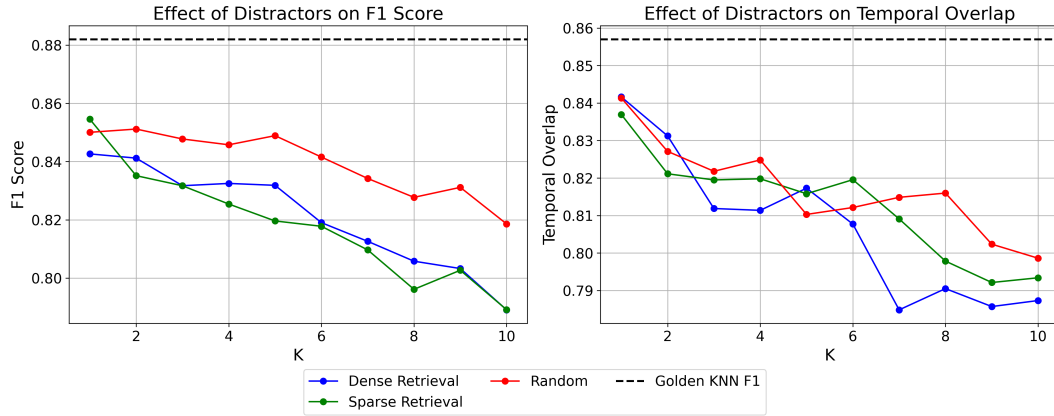


Figure 9: Comparison of F1 Score and Temporal Overlap for answer generation with golden evidence augmented with varying numbers of distractors (k). The dashed line represents the score with golden evidence. Evaluation was conducted on Mistral v0.2

Table 7: Results for Temporal Markers and Time Intervals Across different setups. model: gpt-4o-mini.

Dataset	Setup	Precision	Recall	F1	TO	TJ
TLQA-TM	Closed Book (KNN)	0.449	0.524	0.455	0.589	0.529
	Open Book (BM25@3)	0.703	0.778	0.713	0.820	0.798
	Golden Evidence	0.830	0.938	0.857	0.945	0.934
TLQA-TS	Closed Book (KNN)	0.631	0.485	0.523	0.703	0.589
	Open Book (BM25@3)	0.739	0.656	0.670	0.837	0.719
	Golden Evidence	0.892	0.801	0.823	0.963	0.846

Table 8: Example of incorrect and incomplete answer from TLQA-TM and TLQA-TS.

► Query (TLQA-TM): Which positions did Rob Nicholson hold after serving as Minister of National Defence?	
Expected Answer	Generated Answer
<ul style="list-style-type: none"> Shadow Minister of Justice Shadow Attorney General of Canada: 2015-2019 Minister of Foreign Affairs: 2015 	<ul style="list-style-type: none"> Minister of Foreign Affairs: 2015 Minister of Justice and Attorney General of Canada: 2013-2015 Minister of National Defence: 2013-2013
► Query (TLQA-TS): Which teams did David Beckham play for between 2000 and 2010?	
<ul style="list-style-type: none"> Manchester United F.C.: 2000-2003 Real Madrid CF: 2003-2007 England national football team: 2000-2009 LA Galaxy: 2007-2010 AC Milan (loan): 2009, 2010 	<ul style="list-style-type: none"> Real Madrid: 2003-2007 LA Galaxy: 2007-2012 England national football team: 2000-2009 AC Milan (loan): 2009-2010

5.3 Effects of Distractors

To address RQ 3, we evaluate how different types of distractors impact the performance of the golden evidence based LLM reasoning setup. This would also provide insights into future directions for improving retrieval. We use three different methods for selecting distractors: random selection, retrieving the top k infoboxes using

BM25, or using dense retrieval with all-mini-lm-v2 embeddings with T-S corpora setup. We then append the golden evidence along with these k distractors as evidence for the LLM.

From Figure 9, it can be observed that temporal and list construction performance drops noticeably with the addition of even a single distractor ($k = 1$), indicating that noise affects the model’s ability to extract the correct information to answer the query. This effect

Table 9: Example of refusal errors and incomplete answers.

► Query: List all teams Alan Kirby played to this day. Refusal Error	
Expected Answer	Generated Answer
<ul style="list-style-type: none"> • Aston Villa F.C.: 1996-1998, 1994-1996 • Johnville F.C. (Waterford): 1994 • Republic of Ireland under-21 national football team: 1998 ... 	I apologize, but I couldn't find any information on a person named Alan Kirby playing for any football teams. If you meant someone else or provided incorrect information, please let me know and I'll be happy to help.
► Query: List all political positions Ulf Kristersson held to this day. Incomplete Answer Error	
Expected Answer	Generated Answer
<ul style="list-style-type: none"> • Prime Minister of Sweden: 2022-2024 • Leader of the Moderate Party: 2017-2024 • Leader of the Opposition: 2017-2022 • Minister for Social Security: 2010-2014 • Chairman of the Moderate Youth League: 1988-1992 • Member of the Riksdag: 2014-2024, 1991-2000 	<ul style="list-style-type: none"> • Leader of the Moderate Party: 2017-2024 • Member of the Riksdag: 2006-2021, 2022-2024

becomes more evident as we increase the number of distractors (k). This consistent performance degradation can be attributed to the model's capacity limitations; as more distractors are introduced, it becomes difficult for the model to discern useful information from the correct evidence. Hence, we observe that the model outputs incorrect entities or is unable to output correct time periods due to presence of distracting information. Further, we observe that randomly choosing distractors leads to a less significant decrease in performance compared to distractors retrieved using BM25 or dense retrieval model. We posit that this maybe because the retrieved distractors serve as hard negatives compared to random distractors, rendering it difficult for the LLM to distinguish between golden evidence and the distractors. These results contrast the phenomenon observed recently [9], where the authors observe that adding random noise to contexts improves performance of RAG systems.

5.4 Error Analysis

We perform a detailed error analysis of common errors made by LLMs in our evaluation on the test sets. From Table 9, we observe the case of *refusal errors* where the model is not able to generate an answer in the question due to limitations of parametric knowledge. LLMs are believed to encode world knowledge in their parameters and are believed to perform *approximate retrieval* [20] when posed with new queries. However, we observe that LLMs are unable to provide an answer in some cases as discussed above, either due to limitations in parametric knowledge or inability of the LLM to perform approximate retrieval in closed book setting. We observe that in a retrieval augmented setting for LLMs the **refusal errors** decrease when related evidence is present among retrieved infoboxes.

In Table 9, we also observe *incomplete answer* errors and *incorrect temporal bounds*. We observe that the model only covers two positions, "leader of moderate party" and "member of Riksdag" out of 6 positions held by the individual in question. We posit that incomplete answers could again be an artifact of limited knowledge encoded in model parameters in a closed book setting. We also observe that in many scenarios where the LLM generates incomplete

answers, it is also an artifact of *popularity bias* as the LLM covers popular organizations that are well known and less popular or rare entities/organizational names are left out.

We also observe from Table 9 that the answer "Member of Riksdag" has wrong temporal bounds. We observe that LLM generated time interval has *undershooting errors* where it predicts time period as starting from 2006 when in actuality it was 2014 and similar error for the end time period (2021 vs 2024). it also has an *overshooting problem* where the model predicts 2022 instead of 1991. We observe the *overshooting* and *undershooting* issues for time intervals in closed book and even in open-domain settings, demonstrating that LLMs still lack at accurate temporal understanding.

We also perform error analysis on evaluation subsets TLQA-TS and TLQA-TM as shown in Table 8. We observe from Table 8 that the LLM is unable to detect and reason about the implicit temporal reference made by the temporal marker "after serving as Minister of National Defense" in the question. This results in the LLM generating positions held by "Rob Nicholson" prior to the time period mentioned in the question. Due to incorrect temporal understanding, it also results in missing a position held by this person. We also observe a case of **incorrect temporal bounds**, where the LLM fails to detect the bounds of the time span mentioned in the question and overshoots the time period for the "La Galaxy".

6 Conclusion

We introduce TLQA, an open-domain benchmark where the models are expected to provide a list of possible answers with corresponding time intervals. TLQA primarily tests the temporal understanding and list construction abilities of models. We observe that LLMs believed to encode world knowledge underperform by either predicting incomplete lists or suffer from temporal understanding blindspots resulting in incorrect time intervals or temporal alignment. While we observe that RAG setup improves performance, there exists scope for improvement of tabular retrieval and temporal understanding capabilities of LLMs, providing future research directions.

References

- [1] Samuel Joseph Amouyal, Ohad Rubin, Ori Yoran, Tomer Wolfson, Jonathan Herzig, and Jonathan Berant. 2022. QAMPARI: An Open-domain Question Answering Benchmark for Questions with Many Answers from Multiple Paragraphs. *CoRR abs/2205.12665* (2022). doi:10.48550/ARXIV.2205.12665 arXiv:2205.12665
- [2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bf8ac142f64a-Paper.pdf
- [3] Ricardo Campos, Gaël Dias, Alípio M. Jorge, and Adam Jatowt. 2014. Survey of Temporal Information Retrieval and Related Applications. *ACM Comput. Surv.* 47, 2, Article 15 (aug 2014), 41 pages. doi:10.1145/2619088
- [4] Kaushik Chakrabarti, Zhimin Chen, Siamak Shakeri, Guihong Cao, and Surajit Chaudhuri. 2020. TableQnA: Answering List Intent Queries With Web Tables. *CoRR abs/2001.04828* (2020). arXiv:2001.04828 <https://arxiv.org/abs/2001.04828>
- [5] Jiabao Chen and Yongquan Fan. 2023. Improving temporal question answering using temporal knowledge graph embedding. In *2023 4th International Conference on Computer Engineering and Application (ICCEA)*. 570–575. doi:10.1109/ICCEA58433.2023.10135342
- [6] Wenhui Chen, Xinyi Wang, and William Yang Wang. 2021. A Dataset for Answering Time-Sensitive Questions. arXiv:2108.06314 [cs.CL]
- [7] Ziyang Chen, Jinzhi Liao, and Xiang Zhao. 2023. Multi-granularity Temporal Question Answering over Knowledge Graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 11378–11392. doi:10.18653/v1/2023.acl-long.637
- [8] Ziyang Chen, Xiang Zhao, Jinzhi Liao, Xinyi Li, and Evangelos Kanoulas. 2022. Temporal knowledge graph question answering via subgraph reasoning. *Knowledge-Based Systems* 251 (2022), 109134. doi:10.1016/j.knsys.2022.109134
- [9] Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The Power of Noise: Redefining Retrieval for RAG Systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2024, Vol. 17)*. ACM, 719–729. doi:10.1145/3626772.3657834
- [10] 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 (03 2022), 257–273. doi:10.1162/tacl_a_00459 arXiv:https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00459/2004543/tacl_a_00459.pdf
- [11] Simon Gottschalk and Elena Demidova. 2018. EventKG: A Multilingual Event-Centric Temporal Knowledge Graph. arXiv:1804.04526 [cs.CL]
- [12] Bert F. Green, Alice K. Wolf, Carol Chomsky, and Kenneth Laughery. 1961. Baseball: an automatic question-answerer. In *Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference* (Los Angeles, California) (IRE-AIEE-ACM '61 (Western)). Association for Computing Machinery, New York, NY, USA, 219–224. doi:10.1145/1460690.1460714
- [13] Dhruv Gupta and Klaus Berberich. 2014. Identifying Time Intervals of Interest to Queries. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management* (Shanghai, China) (CIKM '14). Association for Computing Machinery, New York, NY, USA, 1835–1838. doi:10.1145/2661829.2661927
- [14] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard de Melo, and Gerhard Weikum. 2011. YAGO2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th International Conference Companion on World Wide Web* (Hyderabad, India) (WWW '11). Association for Computing Machinery, New York, NY, USA, 229–232. doi:10.1145/1963192.1963296
- [15] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *Comput. Surveys* 55, 12 (mar 2023), 1–38. doi:10.1145/3571730
- [16] Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018. TempQuestions: A Benchmark for Temporal Question Answering. In *Companion Proceedings of the The Web Conference 2018* (Lyon, France) (WWW '18). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1057–1062. doi:10.1145/3184558.3191536
- [17] Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. 2021. Complex Temporal Question Answering on Knowledge Graphs. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management* (CIKM '21). ACM. doi:10.1145/3459637.3482416
- [18] Hideo Joho, Adam Jatowt, and Roi Blanco. 2014. NTCIR temporalia: a test collection for temporal information access research. In *Proceedings of the 23rd International Conference on World Wide Web* (Seoul, Korea) (WWW '14 Companion). Association for Computing Machinery, New York, NY, USA, 845–850. doi:10.1145/2567948.2579044
- [19] Hideo Joho, Adam Jatowt, and Roi Blanco. 2013. A survey of temporal web search experience. In *Proceedings of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil) (WWW '13 Companion). Association for Computing Machinery, New York, NY, USA, 1101–1108. doi:10.1145/2487788.2488126
- [20] Subbarao Kambhampati. 2024. Can large language models reason and plan? *Annals of the New York Academy of Sciences* 1534, 1 (March 2024), 15–18. doi:10.1111/nyas.15125
- [21] Nattiya Kanhabua and Avishek Anand. 2016. Temporal Information Retrieval. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Pisa, Italy) (SIGIR '16). Association for Computing Machinery, New York, NY, USA, 1235–1238. doi:10.1145/2911451.2914805
- [22] Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Velocity Yu, Dragomir R. Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. 2022. RealTime QA: What's the Answer Right Now? arXiv abs/2207.13332 (2022). <https://api.semanticscholar.org/CorpusID:251105205>
- [23] Spencer Kelly. 2024. Dumpster-dip: wikipedia dump parser. <https://github.com/spencermountain/dumpster-dip>
- [24] Florian Le Bronnec, Alexandre Véronne, Benjamin Negrevergne, Yann Chevalere, and Alexandre Allauzen. 2024. Exploring Precision and Recall to assess the quality and diversity of LLMs. In *62nd Annual Meeting of the Association for Computational Linguistics*.
- [25] Seongyun Lee, Hyunjae Kim, and Jaewoo Kang. 2023. LIQUID: A Framework for List Question Answering Dataset Generation. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023*, Washington, DC, USA, February 7-14, 2023, Brian Williams, Yiling Chen, and Jennifer Neville (Eds.). AAAI Press, 13014–13024. doi:10.1609/AAAI.V37I11.26529
- [26] Alexandria Leto, Cecilia Aguerrebere, Ishwar Bhati, Ted Willke, Mariano Tepner, and Vy Ai Vo. 2024. Toward Optimal Search and Retrieval for RAG. arXiv:2411.07396 [cs.CL] <https://arxiv.org/abs/2411.07396>
- [27] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What Makes Good In-Context Examples for GPT-3?. In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Association for Computational Linguistics, Dublin, Ireland and Online, 100–114. doi:10.18653/v1/2022.deeLIO-1.10
- [28] Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2018. Stochastic Answer Networks for Machine Reading Comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Iryna Gurevych and Yusuke Miyao (Eds.). Association for Computational Linguistics, Melbourne, Australia, 1694–1704. doi:10.18653/v1/P18-1157
- [29] 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. arXiv preprint arXiv:2302.12297 (2023).
- [30] Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2022. A Survey on Multi-hop Question Answering and Generation. arXiv:2204.09140 [cs.CL]
- [31] Costas Mavromatis, Prasanna Lakkar Subramanyam, Vassilis N. Ioannidis, Soji Adeshina, Phillip R. Howard, Tetiana Grinberg, Nagib Hakim, and George Karypis. 2021. TempoQR: Temporal Question Reasoning over Knowledge Graphs. arXiv:2112.05785 [cs.CL]
- [32] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]
- [33] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. *CoRR abs/1606.05250* (2016). arXiv:1606.05250 <http://arxiv.org/abs/1606.05250>
- [34] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Jian Su, Kevin Duh, and Xavier Carreras (Eds.). Association for Computational Linguistics, Austin, Texas, 2383–2392. doi:10.18653/v1/D16-1264
- [35] Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. 2021. Question Answering Over Temporal Knowledge Graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 6663–6676. doi:10.18653/v1/2021.acl-long.520
- [36] Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple Entity-Centric Questions Challenge Dense Retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 6138–6148.

- [37] Chao Shang, Guangtao Wang, Peng Qi, and Jing Huang. 2022. Improving Time Sensitivity for Question Answering over Temporal Knowledge Graphs. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 8017–8026. doi:10.18653/v1/2022.acl-long.552
- [38] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaeckermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Andrew Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Agüera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle K. Barral, Dale R. Webster, Gregory S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. 2023. Towards Expert-Level Medical Question Answering with Large Language Models. *CoRR* abs/2305.09617 (2023). doi:10.48550/ARXIV.2305.09617 arXiv:2305.09617
- [39] Jungbin Son and Alice Oh. 2023. Time-Aware Representation Learning for Time-Sensitive Question Answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 70–77. doi:10.18653/v1/2023.findings-emnlp.6
- [40] Pat Verga, Sebastian Hofstätter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing Judges with Juries: Evaluating LLM Generations with a Panel of Diverse Models. arXiv:2404.18796 [cs.CL] <https://arxiv.org/abs/2404.18796>
- [41] Jiexin Wang, Adam Jatowt, and Masatoshi Yoshikawa. 2022. ArchivalQA: A Large-scale Benchmark Dataset for Open Domain Question Answering over Historical News Collections. arXiv:2109.03438 [cs.CL]
- [42] Wonjin Yoon, Richard Jackson, Aron Lagerberg, and Jaewoo Kang. 2022. Sequence tagging for biomedical extractive question answering. *Bioinform.* 38, 15 (2022), 3794–3801. doi:10.1093/BIOINFORMATICS/BTAC397
- [43] Chen Zhang, Jiuhe Lin, Xiao Liu, Yuxuan Lai, Yansong Feng, and Dongyan Zhao. 2023. How Many Answers Should I Give? An Empirical Study of Multi-Answer Reading Comprehension. *CoRR* abs/2306.00435 (2023). doi:10.48550/ARXIV.2306.00435 arXiv:2306.00435
- [44] Tingyi Zhang, Jiaan Wang, Zhixu Li, Jianfeng Qu, An Liu, Zhigang Chen, and Hongping Zhi. 2024. MusTQ: A Temporal Knowledge Graph Question Answering Dataset for Multi-Step Temporal Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 11688–11699. doi:10.18653/v1/2024.findings-acl.696
- [45] Zhihan Zhang, Yixin Cao, Chenchen Ye, Yunshan Ma, Lizi Liao, and Tat-Seng Chua. 2024. Analyzing Temporal Complex Events with Large Language Models? A Benchmark towards Temporal, Long Context Understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 1588–1606. doi:10.18653/v1/2024.acl-long.87
- [46] Victor Zhong, Weijia Shi, Wen-tau Yih, and Luke Zettlemoyer. 2022. RoMQA: A Benchmark for Robust, Multi-evidence, Multi-answer Question Answering. *CoRR* abs/2210.14353 (2022). doi:10.48550/ARXIV.2210.14353 arXiv:2210.14353