Question Answering (QA) is an important and well-explored task [6, 13, 9] that helps address precise information needs of the user. Instead of showing a long list of snippets or hyperlinks for a query, QA systems enhance user experience by directly providing an answer to a query with supporting evidence [19]. QA systems are increasingly utilized across various fields, including healthcare, education, and other areas where precise and accurate information is needed [15]. These systems are designed to cater to a wide spectrum of queries, generating answers in forms ranging from single word replies to detailed passages and structured lists.

Example: TLQA

Question: *List all sports teams Robert Lewandowski played for from 2010 to 2020.*
[Answer]: Lech Poznań (2010)
Borussia Dortmund (2010-2014)
FC Bayern Munich (2014-2020)

Figure 1: An example of question for Timeline based QA

The significance of questions that require list based answers: Although, numerous solutions for QA systems have been developed, there has been a lack of focus on queries that require structured, listwise answers. Prior works have largely focused on factoid queries or questions with only simple or only one possible answer [10]. These questions can be solved, often, by retrieving a small set of sentences from a single evidence source. However, solving questions with multiple possible answers or answers over a timeframe require reasoning over multiple sources and structuring the answers accordingly and ensuring completeness of the final output. In this work, question with list based answers are denoted as **ListQA**.

The ListQA task may include inquiries referring to side effects of certain medication, historical events, priority lists or rankings. In such situations, mistakes can carry considerable risk, such as incorrectly including or omitting the name of a symptom in responses to a healthcare-related question. Furthermore, ListQAs are notably frequent in certain domains. Yoon et al. [17] illustrate this within the biomedical field, where they found that **21.9%** of the queries necessitate answers in a list format. In contrast, only **7%** of naturally posed queries are listwise. Moreover, Chakrabarti et al. [2] report that over 10% of Bing's web queries fall into two types, either superlative or list-of-entity. This reflects the prevalence of ListQAs in web browsing. Therefore, the significant occurrence of ListQAs underscores their importance for study. Advancing ListQA technology can enhance user experience by presenting information in a more structured manner, and improve thoroughness of the response by providing comprehensive lists.

The significance of answering Timeline based questions: In this work, we plan to study timeline based ListQA, where questions require a list of answers over a time period. This reflects real-world queries where users are interested in querying historical events or news [16]. These are common queries issued to search engines which are time-sensitive or involve temporal specifiers [3, 7] 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 but also get accurate estimates of the related time periods. While existing works have focused on questions with temporal markers, they were primarily based on Knowledge Graphs (KG) [7, 8, 4, 11, 14] and not text-based, which limits the understanding of temporal evolution and transitions in text. Additionally, these works do not address the aspect of questions which require multiple answers, which is common in real-world scenarios for questions associated with a time-period. Hence, in this work, we focus on text based questions that are time-sensitive in nature and require an ordered list as output according to relevant time periods. An example of a question that requires list based answers along with related time periods is shown in Figure 1. We observe that the player has played for 3 different teams over different time periods. The questions similar to the example help evaluate the capability of generative models to track time evolution of facts and answer accurately based on world knowledge encoded in their weights. **Limitations of existing QA benchmarks**: Unlike typical QA scenarios such as those in SQuAD [12] 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 [18] with appropriate time periods. For instance, to respond to the search "coastal cities in the US that won

the green city award between 2007-2010", the system would have to look in passages about coastal cities in the US [2] as well as cities that won the said award in the specified time frame. This need to not only identify multiple relevant pieces of information, but also to understand how they are interconnected to form a cohesive list increases the complexity of timeline based ListQAs (TLQA). Another challenge that occurs when dealing with TLQA task is maintaining contextual and relational integrity, especially when dealing with composite queries such as "Who composed film scores for movies written by Robert Alan Aurthur between 1950 to 1958?" [1]. In addition, there is the challenge of list ordering and structuring, relevant in the case of ordered lists such as ranks or guides. Furthermore, another issue is handling list-specific ambiguities, which involves determining the breadth or depth of the list when dealing with vague terms like "popular" or "major".

In this project, our goal would be to study the abilities of language models on timeline based list answer generation. This goes beyond traditional QA where answers are of free-form type or extractive or Multiple choice types. Answering Listwise QA has constraints that the system should generate the complete list of answers while also ensuring the time ranges are accurate. We provide a carefully curated QA dataset specifically for this task which comprises in total 5K questions. The dataset is partially derived from tempLLama and an analysis of different types of questions is provided in Table 1. The train,val and test splits and relvant code can be found at https://github.com/VenkteshV/TLQA_NLP_Porject/.

1 Research Questions

- **RQ1**: How does fine-tuned generative models and few-shot prompting of generative models in a closed book QA setting perform on the task of TLQA.
- **RQ2:** How does fine-tuned generative models and few-shot prompting of generative models with retrieved top-k evidence perform on the task of TLQA.
- **RQ2:** Does special handling of temporal markers improve performance of Timeline based QA?

2 Experiments

To answer the above research questions, carry out the following experiments. While you can use exact match for matching entities in generated answer and ground truth answer, measure the matching of entities and timelines separately. Also measure if the answer list is complete. For instance, the generated answer would be penalized according to number of answers left out in the generated list when compared to answers in the ground truth list.

- In the first series of experiments, you will evaluate models on TLQA in a closed book setting. Run generative models like FlanT5-large and Flant5-XL in a few-shot setting by selecting demonstration samples from training set of TLQA. Make predictions on test set of TLQA. If necessary compute is available, you can also evaluate on diverse LLMs like LLama2 and Mistral with the quantized version 7B parameter category of these models. To select few-shot samples, select examples from the training set that are close to each test example by using KNN search in a continuous vector space using any embedding model from sentence-transformers. You can find an example implementation for KNN search used for selecting few-shot samples at https://anonymous.4open.science/r/ICAT-47FC/code/2wikimultihopqa/get_chatgpt_skill_transfer_knn.py. You can reuse the class KnnSearch from here.
- Repeat the same experiments with different number of few-shot samples, k=3,5,7,10 and report your observations and analysis on why the performance changes.
- Fine-tune smaller models like BART or FlanT5 base (clab can be used) and large (if sufficient compute is available) on the train set of given TLQA dataset. Then using the fine-tuned models perform inference on the test set of TLQA in a closed book setting. For fine-tuned FlanT5 models, you can also provide few shot examples and perform another inference run on TLQA for this setup.

- Now we move to a different setup by extending the above models to answer questions when provided with appropriate context. Since the answer generation is conditioned on context, this setup is more similar to Retrieval Augmented generation (RAG) where a retriever is used to get top-k relevant context that can be provided as input the downstream generative model. Use the latest **Wikipedia dump** as collection for retrieving relevant context. You can choose to only use infoboxes from all WIkipedia pages. Use off the shelf retrieval models from huggingface like contriever/tas-b. Retrieve top-1 context and provide the context as input to the non-finetuned versions of generative models from first experiment without few-shot examples. While this can be performed by chaining off the shelf retriever and a generative model, those interested in exploring Langchain for the same can use the notebook at `https://colab.research.google.com/drive/1Eq6Dl2I-gE5tX6KigkS5dJfurJL5ejD8?usp=sharing` for reference.
- Repeat the previous experiments with different values of k=3,5,7,10 and analyze the change in performance.
- Now try incorporating [5] temporal relevance as mentioned in the cited work in addition to cosine similarity based metric to retrieve relevant context and analyze its impact on the final QA performance.

In your report detail the intuition behind the performance changes and also qualitative analysis of few prominent examples. Also detail how the experiments answer the various research questions. An example how how the main results might look like is shown in Table 2.

References

[1] S. J. Amouyal, O. Rubin, O. Yoran, T. Wolfson, J. Herzig, and J. Berant, "QAMPARI: : An open-domain question answering benchmark for questions with many answers from multiple paragraphs," *CoRR*, vol. abs/2205.12665, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2205.12665

[2] K. Chakrabarti, Z. Chen, S. Shakeri, G. Cao, and S. Chaudhuri, "Tableqna: Answering list intent queries with web tables," *CoRR*, vol. abs/2001.04828, 2020. [Online]. Available: https://arxiv.org/abs/2001.04828

[3] W. Chen, X. Wang, and W. Y. Wang, "A dataset for answering time-sensitive questions," 2021.

[4] Z. Chen, J. Liao, and X. Zhao, "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)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 11 378–11 392. [Online]. Available: https://aclanthology.org/2023.acl-long.637

[5] A. Gade and J. Jetcheva, "It's about time: Incorporating temporality in retrieval augmented language models," 2024.

[6] B. F. Green, A. K. Wolf, C. Chomsky, and K. Laughery, "Baseball: an automatic question-answerer," in *Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference*, ser. IRE-AIEE-ACM '61 (Western). New York, NY, USA: Association for Computing Machinery, 1961, p. 219–224. [Online]. Available: https://doi.org/10.1145/1460690.1460714

[7] Z. Jia, A. Abujabal, R. Saha Roy, J. Strötgen, and G. Weikum, "Tempquestions: A benchmark for temporal question answering," in *Companion Proceedings of the The Web Conference 2018*, ser. WWW '18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018, p. 1057–1062. [Online]. Available: https://doi.org/10.1145/3184558.3191536

[8] Z. Jia, S. Pramanik, R. Saha Roy, and G. Weikum, "Complex temporal question answering on knowledge graphs," in *Proceedings of the 30th ACM International Conference on Information*

amp; Knowledge Management, ser. CIKM '21. ACM, Oct. 2021. [Online]. Available: http://dx.doi.org/10.1145/3459637.3482416

[9] X. Liu, Y. Shen, K. Duh, and J. Gao, "Stochastic answer networks for machine reading comprehension," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, I. Gurevych and Y. Miyao, Eds. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1694–1704. [Online]. Available: https://aclanthology.org/P18-1157

[10] V. Mavi, A. Jangra, and A. Jatowt, "A survey on multi-hop question answering and generation," 2022.

[11] C. Mavromatis, P. L. Subramanyam, V. N. Ioannidis, S. Adeshina, P. R. Howard, T. Grinberg, N. Hakim, and G. Karypis, "Tempoqr: Temporal question reasoning over knowledge graphs," 2021.

[12] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100, 000+ questions for machine comprehension of text," *CoRR*, vol. abs/1606.05250, 2016. [Online]. Available: http://arxiv.org/abs/1606.05250

[13] ——, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, J. Su, K. Duh, and X. Carreras, Eds. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392. [Online]. Available: https://aclanthology.org/D16-1264

[14] C. Shang, G. Wang, P. Qi, and J. Huang, "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)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8017–8026. [Online]. Available: https://aclanthology.org/2022.acl-long.552

[15] K. Singhal, T. Tu, J. Gottweis, R. Sayres, E. Wulczyn, L. Hou, K. Clark, S. Pfohl, H. Cole-Lewis, D. Neal, M. Schaekermann, A. Wang, M. Amin, S. Lachgar, P. A. Mansfield, S. Prakash, B. Green, E. Dominowska, B. A. y Arcas, N. Tomasev, Y. Liu, R. Wong, C. Semturs, S. S. Mahdavi, J. K. Barral, D. R. Webster, G. S. Corrado, Y. Matias, S. Azizi, A. Karthikesalingam, and V. Natarajan, "Towards expert-level medical question answering with large language models," *CoRR*, vol. abs/2305.09617, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2305.09617

[16] J. Wang, A. Jatowt, and M. Yoshikawa, "Archivalqa: A large-scale benchmark dataset for open domain question answering over historical news collections," 2022.

[17] W. Yoon, R. Jackson, A. Lagerberg, and J. Kang, "Sequence tagging for biomedical extractive question answering," *Bioinform.*, vol. 38, no. 15, pp. 3794–3801, 2022. [Online]. Available: https://doi.org/10.1093/bioinformatics/btac397

[18] C. Zhang, J. Lin, X. Liu, Y. Lai, Y. Feng, and D. Zhao, "How many answers should I give? an empirical study of multi-answer reading comprehension," *CoRR*, vol. abs/2306.00435, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2306.00435

[19] F. Zhu, W. Lei, C. Wang, J. Zheng, S. Poria, and T.-S. Chua, "Retrieving and reading: A comprehensive survey on open-domain question answering," 2021.

Table 1: Comparison of TempLAMA and TLQA Mappings with Wikidata Relations

Wiki ID	**Relation Name**	**Templama Mapping**	**TLQA Mapping**
P54	Member of Sports Team	<subject> plays for <object>.	List all sports teams <subject> played for from <start_year> to <end_year>.
P39	Position Held	<subject> holds the position of <object>.	List all positions <subject> held from <start_year> to <end_year>.
P108	Employer	<subject> works for <object>.	List all employers <subject> worked for from <start_year> to <end_year>.
P102	Political Party	<subject> is a member of the <object>.	List all political parties <subject> was a member of from <start_year> to <end_year>.
P286	Head Coach	<object> is the head coach of <subject>.	List all coaches of <subject> from <start_year> to <end_year>.
P69	Educated At	<subject> attended <object>.	List all educational institutions <subject> attended from <start_year> to <end_year>.
P488	Chairperson	<object> is the chair of <subject>.	List all chairpersons of <subject> from <start_year> to <end_year>.
P6	Head of Government	<object> is the head of the government of <subject>.	List all heads of the government of <subject> from <start_year> to <end_year>.
P127	Owned By	<subject> is owned by <object>.	List all entities that owned <subject> from <start_year> to <end_year>.

Paradigm	ModifiedTempLAMA				ArchivalQA			
	EM	*F*1	*TimeMetric*	*Completeness*	*EM*	*F*1	*TimeMetric*	*Completeness*
Vanilla								
gpt-3.5-turbo-3shot								
mistral-7B-v0.1-3shot								
llama-7B-3shot								
RAG								
gpt-3.5-turbo-3shot								
mistral-7B-v0.1-3shot								
llama-7B-3shot								

Table 2: Table to be filled with results.