Detecting Temporal Ambiguity in Questions

Bhawna Piryani
University of Innsbruck
bhawna.piryani@uibk.ac.at

Abdelrahman Abdallah*
University of Innsbruck
abdelrahman.abdallah@uibk.ac.at

Jamshid Mozafari*
University of Innsbruck
jamshid.mozafari@uibk.ac.at

Adam Jatowt
University of Innsbruck
adam.jatowt@uibk.ac.at

Abstract

Detecting and answering ambiguous questions has been a challenging task in open-domain question answering. Ambiguous questions have different answers depending on their interpretation and can take diverse forms. Temporally ambiguous questions are one of the most common types of such questions. In this paper, we introduce TEMPAMBIQA, a manually annotated temporally ambiguous QA dataset[1] consisting of 8,162 open-domain questions derived from existing datasets. Our annotations focus on capturing temporal ambiguity to study the task of detecting temporally ambiguous questions. We propose a novel approach by using diverse search strategies based on disambiguated versions of the questions. We also introduce and test non-search, competitive baselines for detecting temporal ambiguity using zero-shot and few-shot approaches.

1 Introduction

In the field of open-domain question answering (ODQA) detecting and avoiding ambiguous questions is quite important (Min et al., 2020). Min et al. (2020) found that over 50% of the questions in Google search queries are ambiguous. Current ODQA systems, however, usually operate under an implicit assumption that there is a single correct answer for every question.

Temporal ambiguity, in particular, occurs when a question involves unclear or unspecified time frames, leading to different answers depending on the assumed temporal context (Jia et al., 2024). Temporal ambiguity is then a specific type of ambiguity where the interpretation of a question depends on the time frame being referred to. For example, the question "Who was the president of NBC Universal?" is temporally ambiguous as the answer depends on the specific time frame. Temporal ambiguity poses unique challenges for QA systems, as these need to understand the temporal context of a question to provide the correct answer (Harabagiu and Bejan, 2005). However, its detection should be useful for improving temporal IR and QA systems (Kawai et al., 2010; Joho et al., 2015; Jia et al., 2024).

The objective of our work is to stimulate the design of ODQA systems that are able to distinguish between temporally ambiguous and non-ambiguous questions. This involves understanding the temporal context of a question, a challenge that is particularly prevalent in open-domain question-answering systems. Our work, therefore, extends the existing research in the field of ambiguity in open-domain questions, with a specific focus on temporal aspects.

To foster the research in temporal ambiguity detection, we construct a dataset called TEMPAMBIQA having 8,162 questions (3,879 Ambiguous Questions and 4,283 Unambiguous Questions) using an open-domain version of SituatedQA (Zhang and Choi, 2021), ArchivalQA (Wang et al., 2022) and AmbigQA (Min et al., 2020) datasets. For each question, we manually identify its temporal context, and label the question as ambiguous or unambiguous. If a question has multiple answers due to temporal ambiguity, we annotate the question as ambiguous. For example, consider the question "Who won the World Cup when it was held in South America?". This is a temporally ambiguous question because the answer depends on the specific time frame. A person reading this question would note that it could refer to different time frames (1970, 1978, 1986, etc.), each with a different winner. The question could have multiple answers depending on temporal context.

We establish initial performance benchmarks on TEMPAMBIQA by introducing a comprehensive set of strong baseline methods. (1) Zero-Shot Ques-

*Equal contribution.

[1]The dataset is freely available at https://github.com/DataScienceUIBK/TempAmbiQA.

Findings of the Association for Computational Linguistics: EACL 2024, pages 9620–9634

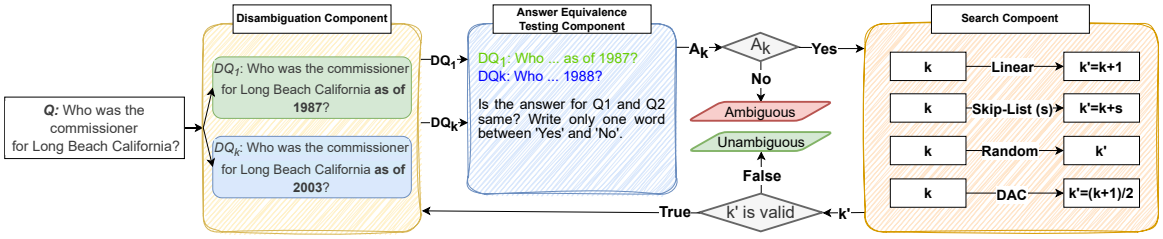

Figure 1: Overview of different search strategies for detecting temporally ambiguous Questions. The Disambiguation Component generates questions DQ_1 and DQ_k, referred to as Q1 and Q2 in the prompts, respectively. The Answer Equivalence Testing Component compares them, classifying Q as temporally ambiguous if the answer equivalence (A_k) is "No". If "Yes", the search proceeds to find the next valid year k' within the defined time range, generating the next disambiguation question $DQ_{k'}$ to continue the classification process. If no valid k' is found, the question Q is classified as temporally unambiguous. A valid year k' is the one that falls within the specified time range (e.g., 2000-2024).

tion Classification enables the model to classify questions into categories for which it has not seen any examples, thereby increasing its ability to handle novel temporal contexts. (2) Few-shot Question Classification allows the model to learn from a small number of examples, making it adaptable to a variety of temporal contexts with limited training data. (3) Fine-Tuned BERT (Devlin et al., 2019) base Model for Question Classification: We fine-tune a BERT base model specifically for the task of question classification.

Our contributions can be summarised as follows:

- We present and release TEMPAMBIQA dataset.
- We propose and test diverse search strategies to assess the temporal ambiguity of questions.

2 Related Work

The development of datasets that explicitly consider temporal and ambiguous aspects remains limited. While datasets like AmbigQA (Min et al., 2020) and others (Kwiatkowski et al., 2019) incorporate certain elements of ambiguity, they do not focus on the temporal dimension. AmbigQA highlights the prevalence of ambiguities in natural questions and studies the disambiguation of questions to address them. However, it does not focus specifically on temporal ambiguities.

ArchivalQA (Wang et al., 2022) is a temporal ODQA dataset featuring questions derived from the New York Times news collection (Sandhaus, 2008) spanning 1987-2007. Zhang and Choi (2021) propose a QA dataset that focuses on temporal and geographical context-dependent questions. Other works (Xu et al., 2019; Aliannejadi et al., 2019; Zamani et al., 2020) use clarification questions to handle ambiguities, but these approaches primarily refine user intents rather than directly resolving temporal confusion. Self-calibrating models (Kumar and Sarawagi, 2019; Cole et al., 2023) have been proposed to estimate confidence and handle ambiguities, but these methods have not been specifically tailored to address the temporal aspect in QA, indicating a potential area for future research.

To the best of our knowledge, we are the first to introduce a dataset specifically focused on temporally ambiguous questions and propose approaches for detecting temporal ambiguity in questions.

3 Data Collection

TEMPAMBIQA is constructed by incorporating questions from different QA datasets, ArchivalQA, SituatedQA, and AmbigQA. To create TEMPAMBIQA, we combined a subset of ArchivalQA that were designated by its authors as containing temporal ambiguous questions, a test set of time-dependent questions from SituatedQA, and the development set of AmbigQA. We then manually labeled all the questions as temporally ambiguous or unambiguous by carefully checking if answers vary over time or the questions can be reformulated into multiple unambiguous variants. This process resulted in a dataset comprising in total 8,162 questions, with 3,879 labeled as ambiguous and 4,283 as unambiguous. Further details about the statistics of TEMPAMBIQA dataset and its few examples can be found in Appendix A.

4 Search Methods

Answers to temporally ambiguous questions change based on the time period they refer to. For

Model	Parameters	Linear Search				Skip List (2) Search				Random (5) Search				Divide And Conquer			
		ACC	PR	RC	F1	ACC	PR	RC	F1	ACC	PR	RC	F1	ACC	PR	RC	F1
T5	770m	0.524	0.499	0.326	0.394	0.524	0.499	0.304	0.378	0.531	0.512	0.281	0.363	0.524	0.499	0.326	0.394
	3b	0.52	0.309	0.007	0.015	0.522	0.324	0.006	0.012	0.521	0.282	0.005	0.01	0.52	0.309	0.007	0.015
LLaMa3	8b	0.509	0.492	0.968	0.652	0.519	0.497	0.958	0.654	0.528	0.502	0.94	0.654	0.509	0.492	**0.968**	0.652
	70b	0.638	0.584	0.821	0.683	0.641	0.589	0.807	0.681	0.643	0.594	0.784	0.676	0.638	0.584	0.821	0.683
Qwen	72b	0.593	0.541	0.941	0.687	0.6	0.547	0.933	0.689	0.61	0.554	0.922	0.692	0.593	0.541	0.941	0.687
	110b	0.641	0.581	0.873	0.698	0.647	0.587	0.864	**0.699**	**0.652**	**0.594**	0.848	0.698	0.641	0.581	0.873	0.698
Mixtral	7b	0.561	0.521	0.95	0.673	0.571	0.528	0.94	0.676	0.58	0.534	0.92	0.675	0.561	0.521	0.95	0.673
	22b	0.628	0.57	0.886	0.693	0.634	0.576	0.875	0.694	0.646	0.587	0.86	0.698	0.628	0.57	0.886	0.693

Table 1: Performance of different LLMs on TEMPAMBIQA dataset using different search approaches. Underlined values represent the best performance across all LLMs for a particular search strategy. Values that are bold indicate the best-performing search strategy across all LLMs.

example, *"Who was the first female Governor of India?"* has a single answer, making it unambiguous. But, *"Who was a Governor of India?"* is temporally ambiguous, as the answer varies over time. To detect such ambiguity, we employ various search strategies by explicitly specifying the relevant year in the question.

To identify a question as ambiguous, we need to find at least two different answers for the same question, each from a different time frame. We do this by disambiguating the question by adding a specific year such as "as of 2001?" at the end of the question. For the TEMPAMBIQA, we use two time frames based on the original datasets. For questions from ArchivalQA, the time frame spans from 1987-2007 to match the news collection period. For questions from SituatedQA and AmbigQA, which are not tied to a specific time range, we set a time frame to span 2000 - 2024. Figure 1 provides an overview of the framework based on a search method used for the classification. The framework consists of three main components described below: Disambiguation Component, Answer Equivalence Testing Component, and Search Component.

4.1 Disambiguation Component

Given a question Q, we generate a disambiguated question DQ_k by appending a specific year from the time frame $T = \{t_1, t_2, \ldots, t_k\}$ to Q.

4.2 Answer Equivalence Testing Component

For each pair of disambiguated questions DQ_1 and DQ_k, we generate answers A_1 and A_k. We then check for semantic equivalence between these answers. If the answers differ, the question is marked as ambiguous. Otherwise, it is passed on to the Search Component. Formally, we compute:

$$AE(A_1, A_k) = \begin{cases} \text{Yes} & \text{if } A_1 = A_k \\ \text{No} & \text{if } A_1 \neq A_k \end{cases}$$

4.3 Search Component

The search component employs various search strategies to efficiently determine temporal ambiguity by finding a pair of differing answers.

Linear Search: The naive approach involves sequentially disambiguating the question for each year in the time frame T:$\{t_1, t_2, \ldots, t_k\}$. Answers for each pair of disambiguated questions DQ_1 and DQ_k are then compared. However, such a linear search method is impractical as it requires comparing answers for every single year with answer for DQ_1, resulting in a large number of comparisons.

Skip-List Search: To improve the search efficiency, we employ a different search approach, i.e., Skip-List search strategy. Unlike the linear search, where the answer to disambiguated question for each consecutive year is compared with DQ_1, the skip-list search compares answers for years at s intervals. For example, in the Skip-list 2 approach, we compare answers for every second year. We implement three different skip-list strategies: Skip-List (2), Skip-List (5), and Skip-List (10), each increasing the gap size of 2, 5, and 10 years. This method reduces the number of comparisons while still effectively identifying temporal ambiguity.

Random Search: Another search strategy we consider is random search. In this approach, we randomly select years from the time frame T :$\{t_1, t_2, \ldots, t_k\}$ and compare the answer for the disambiguated question from these randomly chosen years with the answer for the first disambiguated question DQ_1. We apply different strategies for random search such as finding answers for 5 or 10 randomly selected years and then comparing them to classify the question.

Divide and Conquer (DAC): The final strategy we consider is the divide and conquer approach.

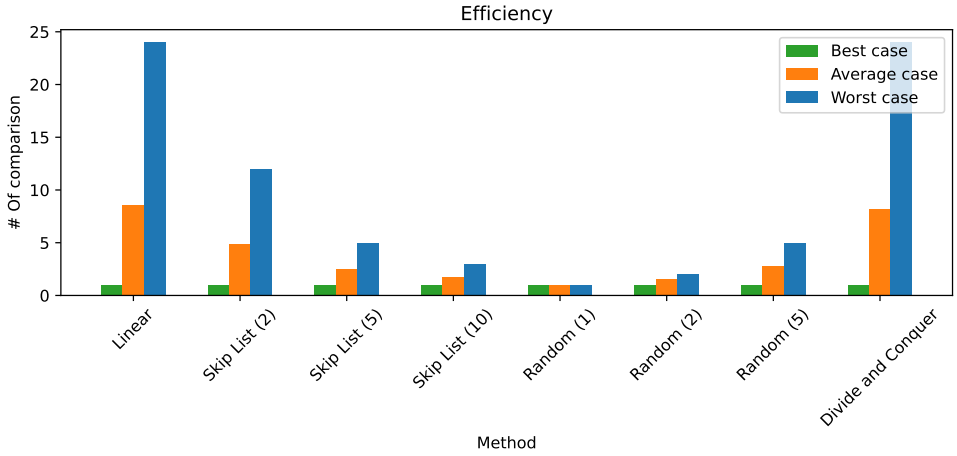

Figure 2: Efficiency of various search strategies.

In this strategy, we initially find the answer for the disambiguated question DQ_1.Then, we divide the remaining time frame in half and compare the answer for DQ_1 with the answer for the disambiguated question at the midpoint year of the time frame T. If the answers are the same, we continue the search by dividing the left half of the time frame and comparing the midpoint of this segment with the DQ_1 answer. This process is repeated until we find different answer and identify the question as ambiguous. We apply the divide and conquer strategy in two ways: first, by searching from left to right, or second, by searching from right to left. In the right-to-left strategy, the search starts by comparing answers from the right part of the time frame and then moves to the left half.

5 Experimental Setup

We analyze different search strategies against Zero-Shot, Few-Shot approaches using diverse models, such as T5 (Raffel et al., 2020), LLaMA3 (Touvron et al., 2023), Qwen (Bai et al., 2023), Mixtral (Jiang et al., 2024), GPT-3.5 (Brown et al., 2020), and GPT-4 (Achiam et al., 2023), with different parameter values. Prompts used for the analyzed approaches are given in Appendix C. Additionally, we also analyze search strategies against a fine-tuned BERT (Devlin et al., 2019) model. We utilize standard evaluation metrics, such as Accuracy (ACC), Precision (PR), Recall (RC), and F1 score (F1).

6 Results

Table 1 shows the performance of different search methods using diverse LLMs. The model Qwen-110B consistently outperforms other LLMs in all strategies. In the Linear Search strategy, it achieves an F1 score of 0.698 and a recall of 0.873, but this method is the most computationally intensive. The Skip List-2 Search method improves efficiency and maintains similar performance over different models. The Random (5) Search strategy also shows promising results, suggesting that random sampling can effectively detect temporal ambiguous question. The DAC method achieves the same results as the linear search but with improved efficiency.

In Figure 2 we illustrate the number of comparisons for all search strategies. The best-case scenario occurs when temporal ambiguity is detected after the first comparison, while the worst-case scenario happens when it is identified only after the final comparison. The average case lies between the extremes but tends to be much closer to the best case regarding efficiency. The efficiency in the average case scenario is more similar to the best-case than worst-case scenarios. Comparing Figure 2 and Table 1, we can conclude that Skip List (2) performs best, both in terms of efficiency and F1 score.

Table 2 summarizes the performance of different models in Zero-Shot and Few-Shot settings. In the Zero-Shot setting, GPT-4 demonstrates high recall but moderate precision, indicating a tendency to overestimate ambiguity. LLaMA3-70B offers a more balanced performance with a recall of 0.741 and an F1 score of 0.616. The T5 model is less effective with low precision and recall. In the Few-Shot scenario, the Mixtral-7B model achieves the highest recall and F1 score, showing that minimal targeted training can enhance performance for tem-

Model	Parameters	ACC	PR	RC	F1
Zero-Shot					
T5	770m	0.519	0.442	0.046	0.083
	3b	0.524	0.422	0.005	0.01
LLaMA3	8b	0.507	0.463	0.242	0.318
	70b	0.561	0.527	0.741	0.616
Qwen	72b	0.585	0.607	0.356	0.449
	110b	0.593	0.597	0.445	0.51
Mixtral	7b	0.522	0.498	0.746	0.597
	22b	0.537	0.509	0.715	0.595
GPT 3.5	-	0.551	0.517	0.837	0.639
GPT 4	-	0.48	0.477	**0.998**	0.646
Few-shot					
T5	770m	0.449	0.443	0.617	0.516
	3b	0.506	0.482	0.543	0.511
LLaMA3	8b	0.523	0.498	0.558	0.527
	70b	0.574	0.598	0.314	0.412
Qwen	72b	0.612	**0.634**	0.437	0.517
	110b	0.584	0.556	0.623	0.588
Mixtral	7b	0.525	0.5	0.856	0.631
	22b	0.606	0.631	0.412	0.499
GPT 3.5	-	0.542	0.515	0.629	0.566
GPT 4	-	0.563	0.556	0.397	0.463
Fine-tuned Model					
BERT	110m	0.597	0.69	0.276	0.394
Search Method					
Qwen (Linear)	110b	0.641	0.581	0.873	0.698
Qwen (Skip List (2)	110b	0.647	0.587	0.864	**0.699**
Qwen (Random (5)	110b	**0.652**	0.594	0.848	0.698
Qwen (DAC)	110b	0.641	0.581	0.873	0.698

Table 2: Performance of various LLMs on the TEMPAMBIQA dataset using different approaches. Underlined values represent the best performance across all LLMs for a particular method. Values that are bold indicate the result for the best approach.

poral ambiguity detection. However, Few-Shot settings does not enhance the performance of models with large number of parameters. The Fine-Tuned BERT model achieves high precision but lower recall, highlighting a trade-off between precision and recall in fine-tuned models.

7 Conclusion

In this paper, we address the novel task of detecting temporally ambiguous questions in ODQA by introducing TEMPAMBIQA, a manually annotated dataset of 8,162 temporally ambiguous and unambiguous questions. We perform several experiments using different search strategies to classify the question as temporally ambiguous. Our experiments demonstrate the effectiveness of search-based methods in detecting temporally ambiguous questions. For future work, we plan to explore more dynamic strategies for determining time ranges, such as identifying the timing of recurring events mentioned in the questions and generating disambiguation questions based on those timelines. Additionally, we aim to refine the granularity of the timelines, moving beyond yearly intervals to include finer distinctions, such as months.

Limitations

While our search strategies aid in detecting temporally ambiguous questions, several limitations must be considered:

1. The effectiveness of the search strategy relies heavily on the time frame considered to create unambiguous questions. Since the time frame for a given question is often unknown, detecting ambiguous questions can be challenging.

2. Ambiguity detection depends entirely on the knowledge embedded in the large language model (LLM) used. Larger models might have a better understanding of ambiguous questions compared to smaller models with fewer parameters.

Ethical Considerations and Licensing

Our research leverages the GPT models, licensed under both the OpenAI License and the Apache-2.0 license, as well as the LLaMA models, distributed under Meta's LLAMA 2 Community License Agreement. We strictly adhere to the conditions set forth by these licenses. The datasets we use are sourced from repositories that permit academic use. To encourage ease of use and modification by the research community, we are releasing the artifacts developed during our study under the MIT license. Throughout the project, we have ensured that data handling, model training, and dissemination of results comply with all relevant ethical guidelines and legal requirements.

Acknowledgment

The computational results presented here have been achieved (in part) using the LEO HPC infrastructure of the University of Innsbruck.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 475–484, New York, NY, USA. Association for Computing Machinery.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, and Others. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Jeremy Cole, Michael Zhang, Daniel Gillick, Julian Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. 2023. Selectively answering ambiguous questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 530–543, Singapore. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Sanda Harabagiu and Cosmin Adrian Bejan. 2005. Question answering based on temporal inference. In *Proceedings of the AAAI-2005 workshop on inference for textual question answering*, pages 27–34.

Zhen Jia, Philipp Christmann, and Gerhard Weikum. 2024. Faithful temporal question answering over heterogeneous sources. In *Proceedings of the ACM on Web Conference 2024*, pages 2052–2063.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Hideo Joho, Adam Jatowt, and Roi Blanco. 2015. Temporal information searching behaviour and strategies. *Information Processing & Management*, 51(6):834–850.

Hideki Kawai, Adam Jatowt, Katsumi Tanaka, Kazuo Kunieda, and Keiji Yamada. 2010. Chronoseeker: search engine for future and past events. In *Proceedings of the 4th International Conference on Uniquitous Information Management and Communication*, ICUIMC '10, New York, NY, USA. Association for Computing Machinery.

Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *ArXiv*, abs/1903.00802.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Evan Sandhaus. 2008. The New York Times Annotated Corpus.

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

Jiexin Wang, Adam Jatowt, and Masatoshi Yoshikawa. 2022. Archivalqa: A large-scale benchmark dataset for open-domain question answering over historical news collections. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 3025–3035, New York, NY, USA. Association for Computing Machinery.

Jingjing Xu, Yuechen Wang, Duyu Tang, Nan Duan, Pengcheng Yang, Qi Zeng, Ming Zhou, and Xu Sun. 2019. Asking clarification questions in knowledge-based question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1618–1629, Hong Kong, China. Association for Computational Linguistics.

Hamed Zamani, Gord Lueck, Everest Chen, Rodolfo Quispe, Flint Luu, and Nick Craswell. 2020. Mimics: A large-scale data collection for search clarification. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*,

CIKM '20, page 3189–3196, New York, NY, USA. Association for Computing Machinery.

Michael Zhang and Eunsol Choi. 2021. SituatedQA: Incorporating extra-linguistic contexts into QA. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Dataset Details

Table 3 presents the details about the statistics of TEMPAMBIQA. Table 4 shows a few examples of different questions collected from various datasets and included in TEMPAMBIQA dataset.

	No. of Questions
#Questions	8,162
Ambiguous Questions	3,879
Unambiguous Questions	4,283
Average question length (words)	8.55

Table 3: Basis statistics of TEMPAMBIQA.

Dataset	**Example**	**Label**
ArchivalQA	Q: How many family houses are in Brooklyn?	Ambiguous
	Q: Who bought Soap Opera Digest in 1989?	Unambiguous
	Q: How much gasoline does the Peykan get?	Ambiguous
SituatedQA	Q: What films has Scarlett Johansson been in?	Ambiguous
	Q: Who coaches the Carolina Panthers?	Ambiguous
	Q: What are some ancient Egypt names?	Unambiguous
AmbigQA	Q: Who lived to be the oldest person in the world?	Ambiguous
	Q: Who is the first woman governor in India?	Unambiguous
	Q: What Olympic sport is also known as ice chess?	Unambiguous

Table 4: Examples of different questions collected from different datasets to create TEMPAMBIQA.

B Additional Experimental Results

In this section, we provide a detailed presentation of the results from our experiments across various scenarios. We will explore how different search strategies perform over questions from individual datasets. Table 5 and Table 6 present results for the variations of Skip-List and Random search over TEMPAMBIQA.

C Case Studies

In this section, we delve into several case studies that illustrate the prompts we have chosen, along with examples from our experiments and their respective outcomes. These case studies are designed to illustrate the working of different methods for detecting temporal ambiguity.

Model	Parameters	Skip List (2)				Skip List (5)				Skip List (10)			
		Acc	PR	RC	F1	Acc	PR	RC	F1	Acc	PR	RC	F1
T5	770m	0.524	0.499	0.304	0.378	0.525	0.5	0.279	0.358	0.524	0.498	0.262	0.344
	3b	0.522	0.324	0.006	0.012	0.521	0.299	0.005	0.01	0.523	0.353	0.005	0.009
LLaMA3	8b	0.519	0.497	**0.958**	0.654	0.528	0.502	0.935	0.653	0.537	0.507	0.904	0.65
	70b	0.641	0.589	0.807	0.681	0.647	0.599	0.779	0.677	0.644	0.605	0.72	0.657
Qwen	72b	0.6	0.547	0.933	0.689	0.613	0.556	0.918	0.692	0.626	0.567	0.897	0.695
	110b	0.647	0.587	0.864	0.699	0.66	0.602	0.839	**0.701**	**0.665**	**0.617**	0.778	0.688
Mixtral	7b	0.571	0.528	0.94	0.676	0.583	0.536	0.909	0.675	0.592	0.545	0.849	0.664
	22b	0.634	0.576	0.875	0.694	0.646	0.589	0.848	0.695	0.654	0.603	0.801	0.688

Table 5: Performance of different LLMs on TEMPAMBIQA using a different variations of **Skip-List** search. Underlined values represent the best performance across all LLMs for a particular search strategy. Values that are both bold and underlined indicate the best-performing search strategy across all models.

Model	Parameters	Random (1) Search				Random (2) Search				Random (5) Search			
		ACC	PR	RC	F1	ACC	PR	RC	F1	ACC	PR	RC	F1
T5	770m	0.537	0.529	0.239	0.329	0.535	0.521	0.258	0.345	0.531	0.512	0.281	0.363
	3b	0.524	0.361	0.003	0.007	0.523	0.34	0.004	0.009	0.521	0.282	0.005	0.01
LLaMA3	8b	0.551	0.518	0.807	0.631	0.541	0.51	0.899	0.651	0.528	0.502	**0.94**	0.654
	70b	0.637	0.617	0.623	0.62	0.641	0.603	0.715	0.654	0.643	0.594	0.784	0.676
Qwen	72b	0.635	0.582	0.825	0.682	0.629	0.57	0.898	0.697	0.61	0.554	0.922	0.692
	110b	0.653	0.621	0.693	0.655	**0.663**	**0.612**	0.8	0.693	0.652	0.594	0.848	**0.698**
Mixtral	7b	0.596	0.557	0.727	0.631	0.597	0.548	0.859	0.669	0.58	0.534	0.92	0.675
	22b	0.66	0.625	0.713	0.666	0.657	0.604	0.808	0.691	0.646	0.587	0.86	**0.698**

Table 6: Performance of different LLMs on TEMPAMBIQA using a different variations of **Random** search. Underlined values represent the best performance across all LLMs for a particular search strategy. Values that are both bold and underlined indicate the best-performing search strategy across all models.

Model	Parameters	Skip List (2)				Skip List (5)				Skip List (10)			
		Acc	PR	RC	F1	Acc	PR	RC	F1	Acc	PR	RC	F1
T5	770m	0.500	0.589	0.251	0.352	0.499	0.601	0.221	0.323	0.498	0.610	0.201	0.302
	3b	0.459	0.615	0.003	0.005	0.459	0.600	0.002	0.004	0.459	0.600	0.002	0.004
LLaMA3	8b	0.563	0.555	0.962	0.704	0.564	0.558	0.936	0.699	0.570	0.564	0.901	0.694
	70b	0.639	0.630	0.806	0.707	0.643	0.642	0.773	0.701	0.635	0.650	0.703	0.676
Qwen	72b	0.607	0.584	0.953	0.724	0.614	0.590	0.941	0.725	0.626	0.601	0.919	**0.727**
	110b	0.640	0.617	0.885	**0.727**	0.652	0.631	0.858	**0.727**	**0.657**	**0.652**	0.787	0.713
Mixtral	7b	0.574	0.562	**0.963**	0.71 0	0.579	0.568	0.930	0.705	0.584	0.578	0.862	0.692
	22b	0.620	0.600	0.896	0.719	0.629	0.610	0.868	0.717	0.635	0.626	0.811	0.706

Table 7: Performance of different LLMs on questions from **ArchivalQA** included in TEMPAMBIQA using a different variations of **Skip-List** search. Underlined values represent the best performance across all LLMs for a particular search strategy. Values that are bold indicate the best-performing search strategy across LLMs.

Model	Parameters	Skip List (2)				Skip List (5)				Skip List (10)			
		Acc	PR	RC	F1	Acc	PR	RC	F1	Acc	PR	RC	F1
T5	770m	0.609	0.38	0.593	0.463	0.615	0.383	0.585	0.463	0.613	0.381	0.580	0.460
	3b	0.700	0.262	0.031	0.055	0.700	0.246	0.027	0.048	0.707	0.293	0.023	0.043
LLaMA3	8b	0.400	0.316	0.956	0.475	0.429	0.326	0.944	0.484	0.451	0.334	0.935	0.492
	70b	0.689	0.476	0.929	0.629	0.702	0.487	0.921	0.637	0.707	0.491	0.894	0.634
Qwen	72b	0.614	0.422	**0.969**	0.588	0.644	0.441	0.954	0.604	0.664	0.456	0.944	0.615
	110b	0.711	0.495	0.923	0.645	0.733	0.517	0.910	0.659	0.741	0.526	0.885	0.660
Mixtral	7b	0.573	0.393	0.929	0.553	0.611	0.417	0.921	0.574	0.631	0.428	0.891	0.578
	22b	0.715	0.499	0.948	0.654	0.740	0.524	0.931	0.671	**0.751**	**0.537**	0.916	**0.677**

Table 8: Performance of different LLMs on questions from **SituatedQA** using a different variations of **Skip-List** search. Underlined values represent the best performance across all LLMs for a particular search strategy. Values that are bold indicate the best-performing search strategy across LLMs.

Model	Parameters	Skip List (2)				Skip List (5)				Skip List (10)			
		Acc	PR	RC	F1	Acc	PR	RC	F1	Acc	PR	RC	F1
T5	770m	0.500	0.440	0.320	0.371	0.500	0.439	0.312	0.365	0.499	0.437	0.31	0.363
	3b	0.540	0	0	0	0.54	0	0	0	0.54	0	0	0
LLaMA3	8b	0.492	0.473	**0.932**	0.628	0.514	0.485	0.916	**0.635**	0.508	0.482	0.890	0.625
	70b	0.546	0.505	0.654	0.570	0.550	0.509	0.627	0.562	0.563	0.521	0.612	0.563
Qwen	72b	0.527	0.490	0.727	0.586	0.534	0.495	0.685	0.575	0.541	0.501	0.661	0.57
	110b	0.550	0.509	0.619	0.559	0.553	0.512	0.596	0.551	0.547	0.507	0.562	0.533
Mixtral	7b	0.552	0.509	0.774	0.614	0.548	0.506	0.732	0.599	0.556	0.513	0.696	0.59
	22b	0.546	0.505	0.612	0.553	0.557	0.517	0.575	0.544	**0.569**	**0.529**	0.570	0.549

Table 9: Performance of different LLMs on questions from **AmbigQA** using a different variations of **Skip-List** search. Underlined values represent the best performance across all LLMs for a particular search strategy. Values that are bold indicate the best-performing search strategy across LLMs.

Model	Parameters	Random (1) Search				Random (2) Search				Random (5) Search			
		ACC	PR	RC	F1	ACC	PR	RC	F1	ACC	PR	RC	F1
T5	770m	0.499	0.613	0.204	0.306	0.501	0.608	0.219	0.322	0.501	0.597	0.241	0.343
	3b	0.459	**0.667**	0.001	0.003	0.459	0.625	0.002	0.003	0.459	0.700	0.002	0.005
LLaMA3	8b	0.563	0.568	0.801	0.665	0.567	0.563	0.899	0.692	0.563	0.557	0.948	0.701
	70b	0.615	0.660	0.594	0.625	0.640	0.653	0.717	0.683	0.643	0.638	0.784	0.704
Qwen	72b	0.629	0.613	0.853	0.714	0.625	0.600	0.924	**0.727**	0.614	0.589	**0.950**	**0.727**
	110b	0.647	0.659	0.721	0.689	**0.651**	0.638	0.818	0.717	0.643	0.622	0.867	0.725
Mixtral	7b	0.568	0.579	0.744	0.651	0.582	0.575	0.879	0.695	0.578	0.566	0.945	0.708
	22b	0.627	0.636	0.726	0.678	0.636	0.623	0.831	0.712	0.626	0.606	0.880	0.718

Table 10: Performance of different LLMs on questions from **ArchivalQA** using a different variations of **Random search**. Underlined values represent the best performance across all LLMs for a particular search strategy. Values that are bold indicate the best-performing search strategy across LLMs.

Model	Parameters	Random (1) Search				Random (2) Search				Random (5) Search			
		ACC	PR	RC	F1	ACC	PR	RC	F1	ACC	PR	RC	F1
T5	770m	0.669	0.426	0.474	0.449	0.660	0.419	0.514	0.462	0.643	0.405	0.545	0.464
	3b	0.708	0.222	0.012	0.022	0.704	0.225	0.017	0.032	0.704	0.261	0.023	0.042
LLaMA3	8b	0.503	0.346	0.843	0.491	0.463	0.337	0.919	0.493	0.424	0.323	0.935	0.480
	70b	0.730	0.516	0.810	0.630	0.714	0.498	0.885	0.638	0.699	0.484	0.912	0.632
Qwen	72b	0.703	0.487	0.856	0.621	0.671	0.461	**0.933**	0.617	0.643	0.441	**0.958**	0.604
	110b	0.763	0.558	0.802	0.658	0.743	0.530	0.860	0.655	0.726	0.510	0.912	0.654
Mixtral	7b	0.688	0.469	0.758	0.58	0.657	0.446	0.858	0.587	0.607	0.413	0.908	0.567
	22b	**0.784**	**0.585**	0.821	0.684	0.773	0.562	0.908	0.695	0.748	0.532	0.944	0.680

Table 11: Performance of different LLMs on questions from **SituatedQA** using a different variations of **Random search**. Underlined values represent the best performance across all LLMs for a particular search strategy. Values that are bold indicate the best-performing search strategy across LLMs.

Model	Parameters	Random (1) Search				Random (2) Search				Random (5) Search			
		ACC	PR	RC	F1	ACC	PR	RC	F1	ACC	PR	RC	F1
T5	770m	0.506	0.426	0.213	0.284	0.507	0.437	0.244	0.313	0.501	0.429	0.255	0.32
	3b	0.54	0	0	0	0.54	0	0	0	0.54	0	0	0
LLaMA3	8b	0.524	0.489	0.787	0.604	0.523	0.49	0.879	0.629	0.51	0.483	**0.921**	**0.634**
	70b	0.572	0.536	0.533	0.534	0.554	0.514	0.593	0.551	0.55	0.508	0.64	0.567
Qwen	72b	0.55	0.509	0.575	0.54	0.542	0.502	0.646	0.565	0.533	0.494	0.693	0.577
	110b	0.539	0.499	0.444	0.469	0.55	0.51	0.533	0.521	0.554	0.513	0.609	0.557
Mixtral	7b	0.574	0.534	0.58	0.556	0.562	0.518	0.664	0.582	0.562	0.516	0.748	0.611
	22b	**0.579**	**0.549**	0.467	0.505	0.563	0.525	0.53	0.527	0.566	0.525	0.596	0.558

Table 12: Performance of different LLMs on questions from **AmbigQA** using a different variations of **Random search**. Underlined values represent the best performance across all LLMs for a particular search strategy. Values that are bold indicate the best-performing search strategy across LLMs.

Model	Parameters	Linear Search				Skip List (2)				Random (5) Search				Divide and Conquer			
		ACC	PR	RC	F1	ACC	PR	RC	F1	ACC	PR	RC	F1	ACC	PR	RC	F1
T5	770m	0.500	0.579	0.279	0.377	0.5	0.589	0.251	0.352	0.501	0.597	0.241	0.343	0.5	0.579	0.279	0.377
	3b	0.46	0.667	0.003	0.007	0.459	0.615	0.003	0.005	0.459	**0.700**	0.002	0.005	0.46	0.667	0.003	0.007
LLaMA3	8b	0.557	0.552	**0.970**	0.703	0.563	0.555	0.962	0.704	0.563	0.557	0.948	0.701	0.557	0.552	**0.970**	0.703
	70b	0.636	0.624	0.821	0.709	0.639	0.630	0.806	0.707	0.643	0.638	0.784	0.704	0.636	0.624	0.821	0.709
Qwen	72b	0.603	0.581	0.958	0.723	0.607	0.584	0.953	0.724	0.614	0.589	0.950	**0.727**	0.603	0.581	0.958	0.723
	110b	0.635	0.612	0.893	0.726	0.640	0.617	0.885	**0.727**	**0.643**	0.622	0.867	0.725	0.635	0.612	0.893	0.726
Mixtral	7b	0.566	0.557	0.969	0.707	0.574	0.562	0.963	0.71	0.578	0.566	0.945	0.708	0.566	0.557	0.969	0.707
	22b	0.615	0.595	0.908	0.719	0.62	0.6	0.896	0.719	0.626	0.606	0.88	0.718	0.615	0.595	0.908	0.719

Table 13: Performance of different LLMs on questions from **ArchivalQA** dataset using different search approaches. Underlined values represent the best performance across all LLMs for a particular search strategy. Values that are bold indicate the best-performing search strategy across LLMs.

Model	Parameters	Linear Search				Skip List (2)				Random (5) Search				Divide and Conquer			
		ACC	PR	RC	F1	ACC	PR	RC	F1	ACC	PR	RC	F1	ACC	PR	RC	F1
T5	770m	0.608	0.379	0.595	0.463	0.613	0.381	0.58	0.46	0.66	0.419	0.514	0.462	0.608	0.379	0.595	0.463
	3b	0.694	0.241	0.036	0.063	0.707	0.293	0.023	0.043	0.704	0.225	0.017	0.032	0.694	0.241	0.036	0.063
LLaMA3	8b	0.378	0.309	0.964	0.468	0.451	0.334	0.935	0.492	0.463	0.337	0.919	0.493	0.378	0.309	0.964	0.468
	70b	0.682	0.470	0.931	0.625	0.707	0.491	0.894	0.634	0.714	0.498	0.885	0.638	0.682	0.47	0.931	0.625
Qwen	72b	0.592	0.409	0.983	0.578	0.664	0.456	**0.944**	0.615	0.671	0.461	0.933	0.617	0.592	0.409	0.983	0.578
	110b	0.7	0.486	0.933	0.639	0.741	0.526	0.885	0.66	0.743	0.53	0.86	0.655	0.700	0.486	0.933	0.639
Mixtral	7b	0.55	0.383	0.95	0.545	0.631	0.428	0.891	0.578	0.657	0.446	0.858	0.587	0.55	0.383	0.95	0.545
	22b	0.701	0.486	0.95	0.643	0.751	0.537	0.916	0.677	**0.773**	**0.562**	0.908	**0.695**	0.701	0.486	0.95	0.643

Table 14: Performance of different LLMs on questions from **SituatedQA** dataset using different search approaches. Underlined values represent the best performance across all LLMs for a particular search strategy. Values that are bold indicate the best-performing search strategy across LLMs.

Model	Parameters	Linear Search				Skip List (2)				Random (5) Search				Divide and Conquer			
		ACC	PR	RC	F1	ACC	PR	RC	F1	ACC	PR	RC	F1	ACC	PR	RC	F1
T5	770m	0.5	0.441	0.323	0.373	0.5	0.44	0.32	0.371	0.501	0.429	0.255	0.32	0.5	0.441	0.323	0.373
	3b	0.54	0	0	0	0.54	0	0	0	0.54	0	0	0	0.54	0	0	0
LLaMA3	8b	0.483	0.47	**0.953**	**0.629**	0.492	0.473	0.932	0.628	0.51	0.483	0.921	0.634	0.483	0.47	**0.953**	**0.629**
	70b	0.552	0.510	0.675	0.581	0.546	0.505	0.654	0.57	0.55	0.508	0.64	0.567	0.552	0.510	0.675	0.581
Qwen	72b	0.525	0.49	0.751	0.593	0.527	0.49	0.727	0.586	0.533	0.494	0.693	0.577	0.525	0.49	0.751	0.593
	110b	0.542	0.502	0.635	0.561	0.550	0.509	0.619	0.559	0.554	0.513	0.609	0.557	0.542	0.502	0.635	0.561
Mixtral	7b	0.55	0.507	0.798	0.62	0.552	0.509	0.774	0.614	0.562	0.516	0.748	0.611	0.55	0.507	0.798	0.62
	22b	0.547	0.506	0.625	0.559	0.546	0.505	0.612	0.553	**0.566**	**0.525**	0.596	0.558	0.547	0.506	0.625	0.559

Table 15: Performance of different LLMs on questions from **AmbigQA** dataset using different search approaches. Underlined values represent the best performance across all LLMs for a particular search strategy. Values that are bold indicate the best-performing search strategy across LLMs.

Model	Parameters	ACC	PR	RC	F1
		Zero-Shot			
T5	770m	0.467	0.685	0.03	0.057
	3b	0.460	**1**	0.002	0.004
LLaMA3	8b	0.485	0.559	0.229	0.325
	70b	0.570	0.577	0.775	0.661
Qwen	72b	0.560	0.643	0.42	0.508
	110b	0.577	0.644	0.486	0.554
Mixtral	7b	0.549	0.559	0.792	0.655
	22b	0.540	0.553	0.779	0.647
GPT 3.5	-	0.576	0.571	0.873	0.690
GPT 4	-	0.543	0.542	**1**	0.703
		Few-shot			
T5	770m	0.5	0.536	0.563	0.549
	3b	0.511	0.546	0.577	0.561
LLaMA3	8b	0.548	0.589	0.545	0.566
	70b	0.559	0.695	0.330	0.448
Qwen	72b	0.602	0.67	0.522	0.587
	110b	0.606	0.63	0.659	0.644
Mixtral	7b	0.575	0.567	0.910	0.699
	22b	0.593	0.671	0.488	0.565
GPT 3.5	-	0.579	0.612	0.608	0.610
GPT 4	-	0.550	0.638	0.390	0.484
		Fine-tuned Model			
BERT	110m	0.539	0.711	0.252	0.372
		Search Method			
Qwen (Linear)	110b	0.635	0.612	0.893	0.726
Qwen (Skip List)	110b	**0.640**	0.617	0.885	**0.727**
Qwen (Random)	72b	0.614	0.589	0.950	**0.727**
Qwen (DAC)	110b	0.635	0.612	0.893	0.726

Table 16: Performance of various LLMs on questions from **ArchivalQA** dataset using different approaches. Underlined values represent the best performance across all LLMs for a particular method. Values that are bold indicate the result for the best approach.

Model	Parameters	ACC	PR	RC	F1
		Zero-Shot			
T5	770m	0.666	0.29	0.121	0.171
	3b	0.709	0.324	0.021	0.04
LLaMA3	8b	0.577	0.285	0.324	0.303
	70b	0.543	0.346	0.683	0.459
Qwen	72b	0.681	0.363	0.163	0.225
	110b	0.661	0.4	0.386	0.393
Mixtral	7b	0.453	0.307	0.737	0.433
	22b	0.553	0.326	0.539	0.407
GPT 3.5	-	0.487	0.332	0.793	0.468
GPT 4	-	0.296	0.287	**0.998**	0.446
		Few-Shot			
T5	770m	0.297	0.258	0.787	0.389
	3b	0.51	0.173	0.192	0.182
LLaMA3	8b	0.467	0.318	0.762	0.448
	70b	0.635	0.354	0.345	0.35
Qwen	72b	0.68	0.376	0.192	0.254
	110b	0.533	0.313	0.537	0.395
Mixtral	7b	0.411	0.29	0.743	0.417
	22b	0.674	0.363	0.196	0.254
GPT 3.5	-	0.448	0.322	0.856	0.468
GPT 4	-	0.601	0.358	0.507	0.419
		Fine-tuned Model			
BERT	110m	0.764	0.716	0.28	0.403
		Search Method			
Mixtral (Linear)	22b	0.701	0.486	0.95	0.643
Mixtral (Skip list)	22b	0.751	0.537	0.916	0.677
Mixtral (Random)	22b	**0.773**	**0.562**	0.908	**0.695**
Mixtral (DAC)	22b	0.701	0.486	0.95	0.643

Table 17: Performance of various LLMs on the subset of **SituatedQA** dataset using different approaches. Underlined values represent the best performance across all LLMs for a particular method. Values that are bold indicate the result for the best approach.

Model	Parameters	ACC	PR	RC	F1
		Zero-Shot			
T5	770m	0.536	0.472	0.066	0.115
	3b	0.539	0.4	0.005	0.01
LLaMA3	8b	0.495	0.414	0.234	0.299
	70b	0.537	0.498	0.556	0.525
qwen	72b	0.535	0.478	0.113	0.183
	110b	**0.553**	0.538	0.205	0.297
Mixtral	7b	0.496	0.447	0.402	0.423
	22b	0.484	0.441	0.454	0.448
GPT 3.5	-	0.525	0.488	0.619	0.546
GPT 4	-	0.467	0.463	**0.987**	0.630
		Few-Shot			
t5	770m	0.452	0.447	0.803	0.574
	3b	0.459	0.448	0.761	0.564
llama3	8b	0.481	0.429	0.386	0.406
	70b	0.537	0.491	0.144	0.223
qwen	72b	0.534	0.472	0.11	0.179
	110b	0.554	0.518	0.462	0.488
Mixtral	7b	0.441	0.423	0.591	0.493
	22b	0.542	0.512	0.113	0.185
GPT 3.5	-	0.505	0.463	0.480	0.472
GPT 4	-	**0.563**	**0.545**	0.302	0.389
		Fine-tuned Model			
BERT	110m	0.609	0.597	0.462	0.521
		Search Method			
LLaMA3 (Linear)	8b	0.483	0.47	0.953	0.629
LLaMA3 (Skip List)	8b	0.492	0.473	0.932	0.628
LLaMA3 (Random)	8b	0.51	0.483	0.921	**0.634**
LLaMA3 (DAC)	8b	0.483	0.47	0.953	0.629

Table 18: Performance of various LLMs on questions from **AmbigQA** dataset using different approaches. Underlined values represent the best performance across all LLMs for a particular method. Values that are bold indicate the result for the best approach.

Zero-Shot setting	**Actual Label**
Is the following question ambiguous? Just give answer as 'YES' or 'NO'. Question: Who won the last olympic men's hockey? Answer: YES	Ambiguous
Is the following question ambiguous? Just give answer as 'YES' or 'NO'. Question: Who is the first woman governor in india? Answer: NO	Unambiguous

Table 19: Case study of detecting temporally ambiguous questions in **Zero-Shot setting**. Words in blue indicate the correct answer. Words in red indicate the answer by LLM.

Few-Shot setting
Is the following question ambiguous? Just give answer as 'YES' or 'NO'. Question: How many dominant racecars did Harvick drive? Answer: No Is the following question ambiguous? Just give answer as 'YES' or 'NO'. Question: Where is the Maya Hieroglyphics Conference held? Answer: Yes Is the following question ambiguous? Just give answer as 'YES' or 'NO'. Question: What is Brian Deletka's job title? Answer: Yes Is the following question ambiguous? Just give answer as 'YES' or 'NO'. Question: What is Jalal Talabani the leader of? Answer: No Is the following question ambiguous? Just give answer as 'YES' or 'NO'. Question: Who is Blankenship's White House adviser? Answer: No Is the following question ambiguous? Just give answer as 'YES' or 'NO'. Question: Where was the gas giveaway in Hackensack? Answer: Yes Is the following question ambiguous? Just give answer as 'YES' or 'NO'. Question: Who is the first woman governor in india? Answer: NO

Table 20: Case study of detecting temporally ambiguous questions in **Few-Shot setting**. The answer by LLM is indicated in red. The true label for the example question is *Unambiguous*.

Q1: who is president of india in present time as of 2000?
Q2: who is president of india in present time as of 2011?
Is the answer for Q1 and Q2 same? Write only one word between 'Yes' and 'No'.
Answer: No

Q1: Who issued ashwamedha coins after performing ashvamedha sacrifice as of 2000?
Q2: Who issued ashwamedha coins after performing ashvamedha sacrifice as of 2001?
Is the answer for Q1 and Q2 same? Write only one word between 'Yes' and 'No'.
Answer: Yes

Q1: who has the oldest team in the nba as of 2000?
Q2: who has the oldest team in the nba as of 2001?
Is the answer for Q1 and Q2 same? Write only one word between 'Yes' and 'No'.
Answer: YES

.....
.....

Q1: who is president of india in present time as of 2000?
Q2: who is president of india in present time as of 2011?
Is the answer for Q1 and Q2 same? Write only one word between 'Yes' and 'No'.
Answer: No

Q1: Who issued ashwamedha coins after performing ashvamedha sacrifice as of 2000?
Q2: Who issued ashwamedha coins after performing ashvamedha sacrifice as of 2001?
Is the answer for Q1 and Q2 same? Write only one word between 'Yes' and 'No'.
Answer: Yes

Q1: who has the oldest team in the nba as of 2000?
Q2: who has the oldest team in the nba as of 2022?
Is the answer for Q1 and Q2 same? Write only one word between 'Yes' and 'No'.
Answer: NO

.....
.....

Q1: who is president of india in present time as of 2000?
Q2: who is president of india in present time as of 2011?
Is the answer for Q1 and Q2 same? Write only one word between 'Yes' and 'No'.
Answer: No

Q1: Who issued ashwamedha coins after performing ashvamedha sacrifice as of 2000?
Q2: Who issued ashwamedha coins after performing ashvamedha sacrifice as of 2001?
Is the answer for Q1 and Q2 same? Write only one word between 'Yes' and 'No'.
Answer: Yes

Q1: who has the oldest team in the nba as of 2000?
Q2: who has the oldest team in the nba as of 2024?
Is the answer for Q1 and Q2 same? Write only one word between 'Yes' and 'No'.
Answer: NO

Table 21: Case study of computing answer equivalence between two questions for various search strategies using Few-Shot setting. Yes indicates the answers for Q1 and Q2 are same whereas NO indicates the answers for Q1 and Q2 are different.

Few-Shot Linear Search

Question : who has the oldest team in the NBA?

Q1: who is president of india in present time as of 2000?
Q2: who is president of india in present time as of 2011?
Is the answer for Q1 and Q2 same? Write only one word between 'Yes' and 'No'.
Answer: No

Q1: Who issued ashwamedha coins after performing ashvamedha sacrifice as of 2000?
Q2: Who issued ashwamedha coins after performing ashvamedha sacrifice as of 2001?
Is the answer for Q1 and Q2 same? Write only one word between 'Yes' and 'No'.
Answer: Yes

Q1: *who has the oldest team in the nba as of 2000?*
Q2: *Question*
Is the answer for Q1 and Q2 same? Write only one word between 'Yes' and 'No'.
Answer: *Answer*

Diambiguated Questions	**Answers**
who has the oldest team in the nba as of 2001?	Yes
who has the oldest team in the nba as of 2002?	Yes
who has the oldest team in the nba as of 2003?	Yes
who has the oldest team in the nba as of 2004?	Yes
who has the oldest team in the nba as of 2005?	Yes
who has the oldest team in the nba as of 2006?	Yes
who has the oldest team in the nba as of 2007?	Yes
who has the oldest team in the nba as of 2008?	Yes
who has the oldest team in the nba as of 2009?	Yes
who has the oldest team in the nba as of 2010?	Yes
who has the oldest team in the nba as of 2011?	Yes
who has the oldest team in the nba as of 2012?	Yes
who has the oldest team in the nba as of 2013?	Yes
who has the oldest team in the nba as of 2014?	Yes
who has the oldest team in the nba as of 2015?	Yes
who has the oldest team in the nba as of 2016?	Yes
who has the oldest team in the nba as of 2017?	Yes
who has the oldest team in the nba as of 2018?	Yes
who has the oldest team in the nba as of 2019?	Yes
who has the oldest team in the nba as of 2020?	Yes
who has the oldest team in the nba as of 2021?	Yes
who has the oldest team in the nba as of 2022?	NO
who has the oldest team in the nba as of 2023?	NO
who has the oldest team in the nba as of 2024?	NO

Table 22: Case study for detecting temporally ambiguous questions using different search strategies. Yes indicates that the answers for Q1 and Q2 are same whereas No indicates the answers for Q1 and Q2 are different. The table here shows the answer equivalence of the corresponding question with Q1 mentioned in the prompt. In the case of linear search, the number of comparisons to classify the question as ambiguous will be 22, whereas for Skip List (2), it will be 11.