Can LLMs Help Encoder Models Maintain Both High Accuracy and Consistency in Temporal Relation Classification?

Adiel Meir[1], **Kfir Bar**[1]

[1]Efi Arazi School of Computer Science, Reichman University, Israel

Correspondence: kfir.bar@runi.ac.il

Abstract

Temporal relation classification (TRC) demands both accuracy and temporal consistency in event timeline extraction. Encoder-based models achieve high accuracy but introduce inconsistencies because they rely on pairwise classification, while LLMs leverage global context to generate temporal graphs, improving consistency at the cost of accuracy. We assess LLM prompting strategies for TRC and their effectiveness in assisting encoder models with cycle resolution. Results show that while LLMs improve consistency, they struggle with accuracy and do not outperform a simple confidence-based cycle resolution approach.

1 Introduction

Extracting event timelines from text is a key natural language processing (NLP) task, organizing events chronologically based on their relative occurrence rather than absolute timestamps. A broader definition by (Ocal et al., 2024) describes a timeline as a data structure that arranges events and times in a total order. Timelines have a wide range of practical applications, even when considering events alone. For instance, Bakker et al. (2024) demonstrated how timelines can be used to process government decision letters, extracting and organizing events for improved understanding. Another example is in the medical domain (Sezgin et al., 2023): given a patient's textual medical records—or a collection of such records—it becomes valuable to extract a timeline of relevant medical events to summarize and visualize their journey. Timeline extraction typically involves five steps: (1) *event detection*, identifying relevant events, often treating all verbs as events; (2) *anchoring*, selecting events for comparison; (3) *temporal relation classification (TRC)*, assigning relations to pairs; (4) *graph construction*, combining pairwise relations into a temporal graph; and (5) *timeline extraction*, deriving a timeline from the graph.

Various methods have been proposed for extracting timelines from temporal graphs (Mani et al., 2006; Do et al., 2012; Kolomiyets et al., 2012; Xue and Zhang, 2018). Recently, Ocal et al. (2024) proposed a method for extracting event timelines from documents annotated with the full TimeML scheme (Saurí et al., 2006), which defines 13 temporal relation types. However, modeling all 13 relations is complex and often results in temporal inconsistencies.

To address the complexity of TimeML's full relation set, several datasets focus on simplified subsets. A widely used resource for temporal relation classification (TRC) is MATRES (Ning et al., 2018b), which reduces the relation types to three deterministic labels—*before*, *after*, and *equals*—along with a *vague* category for uncertain cases. These labels are assigned to a subset of all possible event pairs, a design choice intended to improve annotation consistency and reduce ambiguity.

Despite this simplification, temporal inconsistencies can still arise, particularly with models following a *pairwise* approach—predicting relations independently for each event pair without considering previously predicted labels. For example, a model might predict: A *before* B, B *before* C, and mistakenly, A *after* C. The last relation contradicts the others and creates a temporal cycle, which complicates efforts to derive a consistent, linear event timeline. A real instance of such a cycle, predicted on a MATRES document, is illustrated in Figure 1.

Large language models (LLMs) have achieved state-of-the-art performance across many NLP tasks. However, previous studies have shown that generative LLMs underperform compared to encoder-based models on the TRC task as defined in MATRES (Roccabruna et al., 2024). The advantage of LLMs lies in their ability to encode document-wide information flexibly, which enables them to generate an entire temporal graph in a single step. This capability, recently termed *global*

TRC, offers the potential to reduce temporal inconsistencies by considering all event pairs jointly. Building on prior work in TRC, non-fine-tuned generative LLMs still lag behind smaller supervised models that follow the pairwise approach. However, LLMs' ability to generate the entire temporal graph in a single inference step offers a key advantage: the potential to reduce temporal inconsistencies—a common issue in pairwise models.

Therefore, in this work we make two main contributions:[1]

- We investigate the performance of generative LLMs in extracting temporal graphs. Specifically, we focus on the trade-off between pairwise classification accuracy and the rate of temporal inconsistencies (e.g., cycles) in the resulting graph. Using the MATRES dataset, we explore different approaches to prompt design under various input and output conditions.

- Additionally, we propose a hybrid approach that combines a generative LLM with a standard supervised encoder to improve accuracy while mitigating cycles in the temporal graph.

2 Related work

Temporal relation classification (also know as temporal relation extraction, or TRE) has primarily been addressed using fine-tuned, relatively small encoder-based language models, typically following a pairwise approach in which each event pair is labeled independently.

A key limitation of the pairwise approach is its tendency to produce globally inconsistent outputs. Since these models make independent predictions for each pair of events, they do not take previously predicted labels into account during inference. This lack of global awareness can result in contradictions, such as temporal cycles, which undermine the coherence of the predicted temporal structure and ultimately hinder accurate timeline construction. Despite this limitation, numerous well-established encoder-based methods have been proposed to tackle pairwise TRC. These include approaches that leverage contextualized representations and joint inference strategies to improve local and global consistency (Han et al., 2021; Zhou et al., 2021; Ning et al., 2019; Mathur et al., 2021; Wang et al., 2022, 2023a; Zhang et al., 2022; Zhou et al., 2022; Man et al., 2022; Cohen and Bar, 2023; Niu et al., 2024). While these models have contributed significantly to the field, the challenge of maintaining globally coherent temporal graphs remains a central concern in temporal relation classification.

Early efforts such as Ning et al. (2019) introduced a structured framework for TRC by refining the task with better contextual representations and curated evaluation protocols. Subsequent work expanded this by incorporating global constraints, as in Mathur et al. (2021), which applied joint inference to enforce temporal consistency across event graphs. Similarly, Han et al. (2021) proposed EcoNet, which leveraged event graph structures and global coherence to improve document-level temporal reasoning.

Domain-specific applications have also driven innovation in TRC, particularly in the clinical domain. Zhou et al. (2021) addressed the challenges of TRC in clinical texts, which often involve fragmented or incomplete narratives. Their work demonstrated that specialized models and annotation schemes are necessary to adapt general TRC methods to the clinical setting. (Cohen and Bar, 2023), reframed TRC as a Boolean question answering task. By training a RoBERTa model on Yes/No questions formulated based on the annotation guidelines, they effectively simulated the human annotation process and achieved state-of-the-art results on the MATRES dataset. More recent work by Niu et al. (2024) introduced ContEMPO, a large-scale benchmark for document-level temporal reasoning, combining distant supervision and LLM-based annotation to enhance the breadth and realism of training data.

Several recent studies have also focused on extracting temporal structures beyond pairwise relations. Wang et al. (2022) and Wang et al. (2023a) explored the prediction of document creation times (DCT) and global temporal graphs, respectively, highlighting the importance of temporal anchoring in narrative understanding. Zhang et al. (2022) and Zhou et al. (2022) also tackled full timeline construction, proposing models that jointly identify events and infer their temporal relationships, often integrating external knowledge or reasoning modules.

With LLMs becoming state-of-the-art in many tasks and offering more flexible input handling in a zero-shot setting, recent studies have explored different ways to use them for TRC, both in pairwise

[1]We will release the code upon paper acceptance.

Barack Obama would make...Traditionally, the (intentionally) funny lines by our presidents have had one thing in common: They were self-deprecating. Sure, some presidents have **[EVENT5]used[/EVENT5]** jokes to take jabs at their opponents, but not to the extent of Obama. During his tenure, he has increasingly **[EVENT8]unleashed[/EVENT8]** biting comedic barbs against his critics and political adversaries. These jokes are **[EVENT1000]intended[/EVENT1000]** to do more than simply entertain you. They have an agenda. Obama's humor is often delivered the way a comedian dealing with a heckler would do it. He tries to undermine his opponents with it and get the crowd -- in this case the public -- on his side. I can **[EVENT20]assure[/EVENT20]** you that having a crowd laugh at your critic/heckler is not only effective in dominating them, it's also very satisfying.

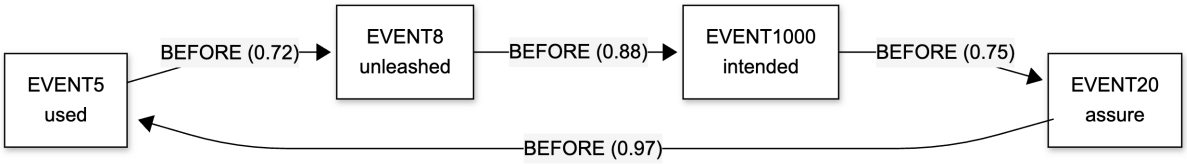

Figure 1: Example of a cycle in a document from the MATRES dataset, mistakenly generated by one of our pairwise encoder models.

and global settings.

Jain et al. (2023) evaluated a variety of LLMs (including standard and code-generation models) across different temporal tasks and prompting strategies (zero-shot, few-shot, CoT). Their comprehensive analysis revealed that while LLMs exhibit proficiency in certain temporal aspects, they face significant challenges in areas requiring reasoning over specific timings and handling complex scenarios involving multiple events. Focusing specifically on the pairwise TRC task, (Roccabruna et al., 2024) investigated if LLMs could supersede established encoder-only models. Evaluating several LLMs with in-context learning and fine-tuning, they found that LLMs generally underperform a strong RoBERTa baseline for this task. Through explainability methods and analysis of word embeddings, they attributed this gap, in part, to differences in pre-training objectives (autoregressive vs. masked language modeling) and how models process input sequences. These studies highlight that while LLMs show promise for broader temporal reasoning, the specific requirements of tasks like pairwise temporal classification may still favor specialized encoder-only architectures or necessitate further research into tailoring LLMs for such fine-grained analysis.

Recent studies have begun exploring the use of zero-shot LLMs for TRC, though most efforts have adhered to the traditional pairwise prediction framework (Yuan et al., 2023; Li et al., 2024; Kougia et al., 2024). A more recent study (Eirew et al., 2025) proposes enhancing global consistency by prompting a strong generative LLM to produce the entire graph of temporal relations in a single step. To address potential contradictions and instability of LLMs in generating consistent output, this approach incorporates a post-processing step based on the linear programming optimization framework introduced by Ning et al. (2018a), which enforces global coherence by resolving inconsistencies in the predicted temporal graph.

Together, these works form a comprehensive foundation for understanding the evolution of TRC, from pairwise classification to global timeline construction, and from specialized supervised models to LLM-based generalization. Building on these efforts, we compare zero-shot LLM accuracy and consistency across prompts and propose a simple, effective cycle-breaking method for encoders while maintaining accuracy.

3 Evaluation of LLMs on TRC

3.1 Extraction Approach

Our timeline extraction approach follows the five-step process outlined in Section 1. Specifically, we work with the MATRES dataset,[2] where all events are defined as verbs. MATRES employs a novel strategy for determining which events should be anchored to a given event. Building on the approximate complete-graph approach introduced in (Naik et al., 2019)—where events are anchored only to those within a predefined surrounding win-

[2]Released under the CC-BY 4.0 license (Ning et al., 2018b), we use the dataset for evaluation as intended by its authors.

dow of sentences—MATRES further refines this by incorporating different types of narrative axes (e.g., opinions, intentions), which impact anchoring decisions. In our work, we build on the MATRES anchoring framework and ask the LLM to merely classify the anchored event pairs according to the MATRES label set: *before*, *after*, *equal*, and *vague*. We explore various approaches to modeling input context length, event marking, yield type, and prompt techniques. Broadly speaking, for a given full document i with k marked events, we use an LLM as a function to predict the corresponding temporal graph. The prompt is structured into three sections: 1) instructions (s_i); 2) the input text ($t_i(e_1, e_2, \ldots, e_k)$), including k marked events to be classified (we mark events as [EVENT1]*eat*[/EVENT1], with the event number taken from the dataset.); and 3) some illustrative input-output examples (f). The output is composed of one or more (m) labels l_i, with each label corresponding to an event pair introduced in the input. Formally, we use the generative LLM as follows:

$$l_1, l_2, \ldots, l_m = LLM[s_i, t_i(e_1, e_2, \ldots, e_k), f]$$

The LLM is expected to return a single label for each event pair formed from the marked events. A subset of these predicted labels is then used as links, along with the marked events, to construct the temporal graph. We noticed that generative LLMs tend to predict the *vague* label more often, likely reflecting their uncertainty. To address this, since LLMs are not instructed to label every pair and can choose which pairs to label, sometimes we removed *vague* from the label set and directed them to predict only *before*, *after*, or *equals*. Furthermore, the *equals* label poses additional challenges for handling in a timeline and is both infrequently annotated and predicted. As a result, we chose to ignore *equals* and *vague* when constructing a temporal graph. Consequently, only the *before* and *after* labels are used as links to form the temporal graph.

We explore variations in prompt design, particularly focusing on the following aspects:

Output Type. Most prior work predicts temporal labels for event pairs individually, a straightforward but inconsistency-prone approach due to its lack of global context. An alternative is predicting the entire temporal graph in one step, leveraging global context for better consistency. We evaluate both approaches—*pairwise* and *graph*. In the graph approach, the model generates labels for all pairs in DOT format (Gansner et al., 2006).

Considered Events. MATRES was selectively annotated, labeling only event pairs within two-sentence paragraphs, leaving many pairs unannotated. To address this, we evaluate accuracy using three approaches. The MATRES approach considers only the originally annotated pairs, using the *pairwise* output type. The *sliding-window* approach expands this by pairing each event with all others in a two-sentence window, shifting one sentence at a time. The *document* approach, applicable only to *graph* output, considers all event pairs but avoids redundancy by marking events and instructing the model to infer non-redundant relations, omitting symmetric and transitive ones to produce a compact graph. In both the *sliding-window* and *document* approaches, event pairs without gold labels but assigned a relation by the model are included for consistency assessment but excluded from accuracy calculations.

Context. The context refers to the portion of text surrounding the events that are provided to the model for classification. We experiment with two context sizes. In the first, referred to as *document*, we provide the entire document to the model. In the second, called *paragraph*, we provide a window of two sentences surrounding the two events in focus. This method is compatible only with the *pairwise* output type.

Prompt Style. We experiment with both zero-shot and few-shot in-context learning. For the *pairwise* output, few-shot learning includes two examples—one *before* and one *after*—randomly selected from the training set. For the *graph* output, we provide a document with marked events and the MATRES-annotated relations in DOT format. Note that this does not fully represent a complete graph, since MATRES provides gold relations only for some of the event pairs. Sample prompts are provided in Appendix A.

3.2 Evaluation Approach

We experiment with all combinations of the prompt aspects mentioned above, using four LLMs: GPT4o, GPT4o-mini, Llama-3.1-8B, and Llama-3.2-3B. We chose these specific models to balance between large and small models, as well as between open and closed sources. Note that not all aspect combinations are possible. For instance,

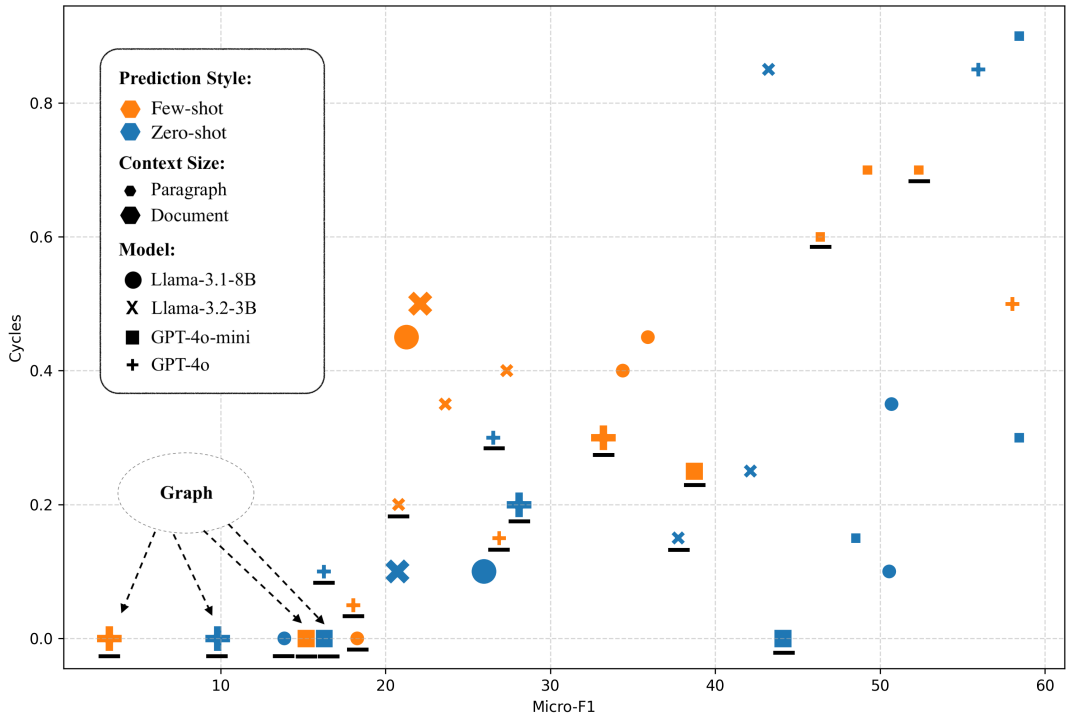

Figure 2: Micro-F1 vs. inconsistency (cycles) for our LLM experiments.

in the *graph* output type, only four combinations exist because both the considered events and context aspects must be set to *document*. Additionally, we evaluate this setting only with OpenAI GPT models, as the task demands stronger reasoning capabilities. Since LLMs are stochastic, we run each instance five times and use majority voting for labeling, following self-consistency (Wang et al., 2023b). Our evaluation balances inconsistency and accuracy: inconsistency is measured by the percentage of test documents with cycles, which prevent timeline extraction, while accuracy is reported as the Micro-F1 score. We evaluate on the MATRES test set (20 documents) and refer to the percentage of cycle-containing documents as the cycle rate.

3.3 Results

Figure 2 offers a high-level overview of our results, highlighting both accuracy and consistency metrics across the different experimental settings. For a comprehensive breakdown, Table 1 presents the full set of results, covering all prompt configurations evaluated with the four large language models used in this study. This includes performance across both pairwise and global prediction modes, as well as the impact of reasoning strategies such as chain-of-thought prompting.

While results are somewhat noisy, we observe a strong correlation between accuracy and cycles ($\rho = 0.64$, $p < .001$, $n = 36$) indicating that higher accuracy is often accompanied by reduced temporal consistency. However, this correlation does not preclude the existence of configurations that simultaneously improve both accuracy and consistency. Indeed, recent work by Eirew et al. (2025) demonstrates that such improvements are achievable under specific settings. We discuss this in more details below.

Predicting *vague* is particularly challenging, as human annotators also struggled with it (Ning et al., 2018b), and it is often represent disagreement between annotators. Therefore, for some experiments, we tested the model both with and without the *vague* label. When included, the model predicted one of four labels (*before*, *after*, *equal*, or *vague*); otherwise, it was limited to three.

Accuracy calculations were based on either the four-label or three-label setting, respectively, following our F1 score definition described above. However, for calculating consistency, that is, the number of documents which introduce cycles, we only consider the *before* and *after* relations. In Figure 2, we indicate experiments that include the *vague* relation by placing a line under each relevant shape. In total, there are seven experiments for which results exist both with and without *vague*.

Model	Output Type	Prompt Style	Considered Event	Context	Vague	Micro-F1	Cycles
Llama-3.1-8B	Pairwise	zero-shot	MARTES	Paragraph	no	50.55	0.10
			Sliding-window	Paragraph	no	50.69	0.35
				Paragraph	yes	13.86	0.00
				Document	no	25.97	0.10
		few-shot	MARTES	Paragraph	no	35.91	0.45
			Sliding-window	Paragraph	no	34.39	0.40
				Paragraph	yes	18.28	0.00
				Document	no	21.27	0.45
Llama-3.2-3B	Pairwise	zero-shot	MARTES	Paragraph	no	42.13	0.25
			Sliding-window	Paragraph	no	43.23	0.85
				Paragraph	yes	37.75	0.15
				Document	no	20.72	0.10
		few-shot	MARTES	Paragraph	no	23.62	0.35
			Sliding-window	Paragraph	no	27.35	0.40
				Paragraph	yes	20.79	0.20
				Document	no	22.1	0.50
GPT-4o-mini	Pairwise	zero-shot	MARTES	Paragraph	no	58.43	0.30
			Sliding-window	Paragraph	no	58.43	0.90
				Paragraph	no	48.51	0.15
				Document	yes	44.09	0.00
		few-shot	MARTES	Paragraph	yes	52.33	0.70
			Sliding-window	Paragraph	no	49.22	0.70
				Paragraph	yes	46.36	0.60
				Document	yes	38.71	0.25
	Graph	zero-shot	Document	Document	yes	16.25	0.00
		few-shot	Document	Document	yes	15.17	0.00
GPT-4o	Pairwise	zero-shot	MARTES	Paragraph	yes	16.25	0.10
			Sliding-window	Paragraph	yes	26.52	0.30
				Paragraph	no	55.94	0.85
				Document	yes	28.08	0.20
		few-shot	MARTES	Paragraph	yes	18.04	0.50
			Sliding-window	Paragraph	yes	26.88	0.15
				Paragraph	no	58.01	0.50
				Document	yes	33.21	0.30
	Graph	zero-shot	Document	Document	yes	9.80	0.00
		few-shot	Document	Document	yes	3.30	0.00

Table 1: Full results of the evaluation of LLMs under different prompting conditions.

Averaging the differences between corresponding experiments, we observe an 18% drop in accuracy when allowing *vague*, but also a 37% reduction in cycled documents. This suggests that when the model can predict *vague*, it does so frequently, which lowers accuracy but also helps break cycles, as only *before* and *after* relations contribute to cycle formation.

The experiments using the *graph* output type are represented by the bubble labeled *graph*. Their accuracy is notably low, suggesting that LLMs struggle to generate the full temporal graph in a single pass using this relatively simple approach supported by previous work (Eirew et al., 2025).

In the *pairwise* (non-*graph*) experiments, the accuracy improves but they introduce more cycles. Another insight is that larger contexts limit accuracy, while smaller ones increase it but add cycles.

Since our goal is timeline generation, we prioritize fewer cycles with reasonable accuracy. The best experiment by this criterion is produced by GPT-4o-mini in a few-shot configuration combined with the *paragraph* context.

3.4 Accuracy vs. Consistency

As reported above, we observe a statistically significant correlation between inconsistency—measured as the number of documents predicted with temporal cycles—and accuracy. In general, our experiments suggest that for non-fine-tuned LLMs, such as those evaluated in this study, higher accuracy often comes at the cost of lower consistency. That is, as the model generates more accurate temporal labels, it also tends to introduce a greater number of contradictory relations. This trade-off appears especially when prompting the model to be more successful (e.g., by providing a more relevant context, or by providing examples) in its predictions, which intuitively aligns with the classic tension between specificity and sensitivity observed in traditional machine learning tasks.

However, it is important to note a fact in our evaluation strategy: consistency is measured over all predicted event-event relations, whereas accuracy is computed with respect to the subset of annotated pairs in the MATRES dataset. Since MATRES does not provide gold labels for event pairs that are more than one sentence apart, but the LLMs output relations for all event pairs, our consistency metric is based on a broader set of predictions than the accuracy metric. This introduces a slight misalignment between the two evaluation dimensions, but we chose to retain this approach to more comprehensively capture the model's global behavior.

While our overall results support the observed trend—that increasing accuracy tends to introduce more inconsistencies—we also find notable exceptions. Certain model and prompt configurations demonstrate improvements in both accuracy and consistency. These cases suggest that, although the trade-off is common, it is not inevitable. Prior work has shown that with careful modeling—such as the use of structured inference or post-hoc consistency enforcement—systems (Eirew et al., 2025) can improve both dimensions simultaneously. Nonetheless, our findings are specific to zero- and few-shot prompting approaches using non-fine-tuned LLMs, and future work may further explore how fine-tuning or additional consistency-aware methods can shift or mitigate this trade-off.

4 Combining LLMs with a Small Encoder

It has already been shown (Roccabruna et al., 2024) that encoder models achieve higher accuracy than generative LLMs on TRC. However, as demonstrated in the previous section, LLMs maintain greater consistency than simple encoders when leveraging global information. Building on this insight, we adopt a hybrid approach, in which a baseline encoder model first predicts temporal relations for all event pairs in a document using the pairwise approach. We then detect simple cycles in the predictions using the NetworkX package (`https://networkx.org/`). A simple cycle is a cycle in a graph with no repeated nodes. As mentioned before, only *before* and *after* relations are considered for cycle detection. Once a cycle is found, we iteratively break it using one of two methods, and the process repeats until no cycles remain in the document. Generally speaking, breaking a simple cycle involves removing a single link from it, aiming to minimize accuracy loss while restoring consistency. We explore two different approaches:
Confidence-Based. This method removes the cycle link with the lowest confidence, determined by the encoder's probability for the predicted label.
LLM-Assisted. This approach prompts a generative LLM to identify the most likely erroneous link in a cycle, leveraging its ability to process detailed input and enhance global consistency. The prompt (Appendix A) provides TRC instructions, requiring the model to identify the most likely error in a document with a cycle of *before* and *after* links. It then presents the full document with marked events and cycle links in DOT format (Gansner et al., 2006).

4.1 Experimental Settings and Results

In addition to MATRES, we use the NarrativeTime (NT) dataset (Rogers et al., 2024) (MIT license), which labels all event pairs in a document rather than just within two-sentence segments. NT also uses seven relations, which are the four of MATRES plus three more *includes*, *is_included*, and *overlap*. The NT test set contains 9 documents (overall, 7,582 relations), 27 training documents (overall, 67,860 relations). We evaluate the two cycle-breaking approaches using an encoder model trained from scratch once on the MATRES training set and once on the NT training set. The model follows the BERT-based (Devlin et al., 2019) (License: Apache 2.0) Entity Marker Entity Start architecture (Baldini Soares et al., 2019) and operates pairwise.

Document	Number of Cycles
CNN_20130321_821	4
CNN_20130322_1003	135
WSJ_20130321_1145	36
WSJ_20130322_159	72
WSJ_20130322_804	59
nyt_20130321_china_pollution	202
nyt_20130321_cyprus	87
nyt_20130321_sarcozy	20
nyt_20130321_women_senate	74

Table 2: Number of simple cycles per each document of the 9 documents that have cycles from the MATRES test set.

Model	**MATRES F1**	**NT F1**
Confidence-Based	74.67	52.28
LLM-Assisted (GPT-4o)	67.03	51.90
LLM-Assisted (GPT-4o-mini)	70.49	51.65

Table 3: Performance of cycle-breaking approaches. Since MATRES is only sparsely annotated, it is hard to know the upper bound for the performance gain only by removing relations to break cycles. For NT, the upper bound is 52.57.

On the MATRES test set, our encoder achieves a micro-F1 score of 80.29%, and on the NT test set, 52.57%. Additionally, 9 of 20 MATRES test documents contain cycles, averaging 76.5 simple cycles per document, while all 9 NT test sets include cycles. Table 2 breaks down the number of simple cycles detected in each document from the MATRES test set. The cycles were detected over the full temporal graph extracted by the base supervised BERT model.

For the LLM-assisted cycle-breaking approach, we experiment with GPT-4o and GPT-4o-mini. All three cycle-breaking approaches successfully resolved all cycles, but accuracy dropped from the original 80.29% (MATRES) and 52.57% (NT). Table 3 summarizes the results. The confidence-based approach significantly outperformed LLM-assisted methods in MATRES and showed less conclusive results in NT, suggesting it was more effective at identifying the correct links to remove.

5 Conclusions

Our study highlights both the promise and the current limitations of using LLMs for timeline extraction through TRC. Across extensive experiments, we observe a recurring inverse relationship between accuracy and consistency: as LLMs are pushed toward higher accuracy through prompt design and reasoning strategies, they tend to generate more globally inconsistent temporal graphs—often resulting in cycles. This trade-off mirrors classic precision–recall tensions in traditional machine learning and highlights the challenge of achieving both local accuracy and global coherence in zero- or few-shot generative settings. We also find that current LLMs struggle to generate complete and accurate temporal graphs in a single pass, even when using compact representations and chain-of-thought reasoning. While supervised encoder-based models remain more accurate on annotated pairs, their pairwise prediction structure inherently introduces global inconsistencies unless followed by post-hoc constraints. Interestingly, when evaluating LLM-generated graphs with existing cycle resolution strategies, a simple confidence-based encoder model remained among the most effective for enforcing consistency—highlighting the value of integrating structured reasoning modules into otherwise generative workflows. Our results motivate future research directions focused on hybrid approaches that combine the strengths of encoder-based models—particularly their structured reasoning and consistency enforcement—with the generalization and contextual understanding of LLMs. We believe that with targeted improvements in prompt engineering, structural guidance, and consistency-aware inference, LLMs can play a central role in advancing temporal relation extraction beyond current limitations.

Limitations

Our study has several limitations. First, our approach Confidence-Based Cycle Breaking relies on confidence scores derived from BERT-based architecture, which may limit the generalization of our conclusions to other architectures or classification strategies. MATRES and NarrativeTime both annotate news articles, so our conclusions may not generalize to other domains. Finally, our experimental evaluation was performed on only four models (two open source and two closed), despite the existence of a broader array of models in the literature.

We see no risks in our work, as we use publicly available datasets as intended and employ LLMs like GPT-4o solely for evaluation. We did not review or filter the datasets for personal information, as both datasets consist solely of publicly available news documents sourced from media outlets.

References

Femke Bakker, Ruben Van Heusden, and Maarten Marx. 2024. Timeline extraction from decision letters using chatgpt. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024)*, pages 24–31.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Omer Cohen and Kfir Bar. 2023. Temporal relation classification using Boolean question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1843–1852, Toronto, Canada. 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.

Quang Do, Wei Lu, and Dan Roth. 2012. Joint inference for event timeline construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 677–687.

Alon Eirew, Kfir Bar, and Ido Dagan. 2025. Beyond pairwise: Global zero-shot temporal graph generation. *Preprint*, arXiv:2502.11114.

Emden Gansner, Eleftherios Koutsofios, and Stephen North. 2006. Drawing graphs with dot.

Rujun Han, Xiang Ren, and Nanyun Peng. 2021. ECONET: Effective continual pretraining of language models for event temporal reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5367–5380, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Raghav Jain, Daivik Sojitra, Arkadeep Acharya, Sriparna Saha, Adam Jatowt, and Sandipan Dandapat. 2023. Do language models have a common sense regarding time? revisiting temporal commonsense reasoning in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6750–6774.

Oleksandr Kolomiyets, Steven Bethard, and Marie Francine Moens. 2012. Extracting narrative timelines as temporal dependency structures. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 88–97.

Vasiliki Kougia, Anastasiia Sedova, Andreas Joseph Stephan, Klim Zaporojets, and Benjamin Roth. 2024. Analysing zero-shot temporal relation extraction on clinical notes using temporal consistency. In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pages 72–84, Bangkok, Thailand. Association for Computational Linguistics.

Xingzuo Li, Kehai Chen, Yunfei Long, and Min Zhang. 2024. Llm with relation classifier for document-level relation extraction. *arXiv preprint arXiv:2408.13889*.

Hieu Man, Nghia Trung Ngo, Linh Ngo Van, and Thien Huu Nguyen. 2022. Selecting optimal context sentences for event-event relation extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 11058–11066.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chungmin Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 753–760.

Puneet Mathur, Rajiv Jain, Franck Dernoncourt, Vlad Morariu, Quan Hung Tran, and Dinesh Manocha. 2021. TIMERS: Document-level temporal relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 524–533, Online. Association for Computational Linguistics.

Aakanksha Naik, Luke Breitfeller, and Carolyn Rose. 2019. TDDiscourse: A dataset for discourse-level temporal ordering of events. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 239–249, Stockholm, Sweden. Association for Computational Linguistics.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018a. Joint reasoning for temporal and causal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2278–2288, Melbourne, Australia. Association for Computational Linguistics.

Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019. An improved neural baseline for temporal relation extraction. 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 6203–6209, Hong Kong, China. Association for Computational Linguistics.

Qiang Ning, Hao Wu, and Dan Roth. 2018b. A multi-axis annotation scheme for event temporal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1318–1328, Melbourne, Australia. Association for Computational Linguistics.

Jingcheng Niu, Saifei Liao, Victoria Ng, Simon De Montigny, and Gerald Penn. 2024. ConTempo: A unified temporally contrastive framework for temporal relation extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1521–1533, Bangkok, Thailand. Association for Computational Linguistics.

Mustafa Ocal, Ning Xie, and Mark Finlayson. 2024. Tlex: An efficient method for extracting exact timelines from timeml temporal graphs. *arXiv preprint arXiv:2406.05265*.

Gabriel Roccabruna, Massimo Rizzoli, and Giuseppe Riccardi. 2024. Will LLMs replace the encoder-only models in temporal relation classification? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20402–20415, Miami, Florida, USA. Association for Computational Linguistics.

Anna Rogers, Marzena Karpinska, Ankita Gupta, Vladislav Lialin, Gregory Smelkov, and Anna Rumshisky. 2024. NarrativeTime: Dense temporal annotation on a timeline. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12053–12073, Torino, Italia. ELRA and ICCL.

Roser Saurí, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. Timeml annotation guidelines. *Version*, 1(1):31.

Emre Sezgin, Syed-Amad Hussain, Steve Rust, and Yungui Huang. 2023. Extracting medical information from free-text and unstructured patient-generated health data using natural language processing methods: feasibility study with real-world data. *JMIR Formative Research*, 7:e43014.

Haoyu Wang, Hongming Zhang, Yuqian Deng, Jacob Gardner, Dan Roth, and Muhao Chen. 2023a. Extracting or guessing? improving faithfulness of event temporal relation extraction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 541–553, Dubrovnik, Croatia. Association for Computational Linguistics.

Liang Wang, Peifeng Li, and Sheng Xu. 2022. DCT-centered temporal relation extraction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2087–2097, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Nianwen Xue and Yuchen Zhang. 2018. Neural ranking models for temporal dependency structure parsing. In *2018 Conference on Empirical Methods in Natural Language Processing*.

Chenhan Yuan, Qianqian Xie, and Sophia Ananiadou. 2023. Zero-shot temporal relation extraction with ChatGPT. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 92–102, Toronto, Canada. Association for Computational Linguistics.

Shuaicheng Zhang, Qiang Ning, and Lifu Huang. 2022. Extracting temporal event relation with syntax-guided graph transformer. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 379–390, Seattle, United States. Association for Computational Linguistics.

Jie Zhou, Shenpo Dong, Hongkui Tu, Xiaodong Wang, and Yong Dou. 2022. RSGT: Relational structure guided temporal relation extraction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2001–2010, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yichao Zhou, Yu Yan, Rujun Han, J Harry Caufield, Kai-Wei Chang, Yizhou Sun, Peipei Ping, and Wei Wang. 2021. Clinical temporal relation extraction with probabilistic soft logic regularization and global inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14647–14655.

A Prompts

Figures 3-7 provide full examples of the prompts used to evaluate the LLMs. In all prompt examples, we use MATRES's four relations, while NarrativeTime prompts follow the same format but include seven relations. Figure 3 presents the prompt for the zero-shot pairwise approach, followed by its few-shot extension in Figure 4. Similarly, Figures 5 and 6 illustrate the prompts for the graph generation approach. Finally, Figure 7 shows the prompt used for breaking cycles with the LLM.

In all examples, we include the *vague* relation, though we also conduct experiments without it.

B Experimental Settings

Both datasets we used in this study, MATRES and NarrativeTime contain documents written in English, and cover news articles. For pairwise model experiments, we evaluate the MATRES approach and sliding-window using Llama-3.1-8B-Instruct-Turbo and Llama-3.2-3B-Instruct-Turbo with Float16 quantization on an NVIDIA GeForce RTX 3090, alongside GPT-4o-mini-2024-07-18 and GPT-4o-2024-08-06, incurring a total cost of approximately $20.

For graph-based model experiments, the same models are used, with OpenAI models costing around $15.

For cycle-breaking models, we train an encoder-based model using the BERT architecture on an NVIDIA GeForce RTX 3090 for five epochs. Training takes approximately 1 hour for MATRES and 7 hours for NarrativeTime. Subsequently, GPT-4o-mini-2024-07-18 and GPT-4o-2024-08-06 are used, with a combined cost of approximately $35.

```
{
  "role": "system",
  "content": "
    Task Overview:
    You are given a text, in which some verbs are uniquely marked by [EVENT#ID]event
        [/EVENT#ID] (e.g., [EVENT1]event1[/EVENT1], [EVENT2]event2[/EVENT2]).
    Your task is to say which of the verbs happened first in a chronological order.
    More specifically, you need to return for each pair of verbs, which is two
        sentence apart,
    a single label out of the listed potential labels:
    before - the first verb happened before the second.
    after - the first verb happened after the second.
    equal - both verbs happened together.
    vague - It is impossible to know based on the context provided

    you should only provide one classification."
},
{
  "role": "user",
  "content": "
    ---
    Text for Analysis:
    Former President Nicolas Sarkozy was [EVENT1]informed[/EVENT1] Thursday that he
        would face a formal investigation into whether he [EVENT3]abused[/EVENT3]
        the frailty of Liliane Bettencourt, 90, the heiress to the L'Oreal fortune
        and France's richest woman, to get funds for his 2007 presidential campaign.
         Mr. Sarkozy has denied accepting illegal campaign funds from Ms.
        Bettencourt, either personally or through his party treasurer at the time,
        Eric Woerth, as alleged by her former butler.
    ---

in one word --> "
}
```

Figure 3: Zero-shot prompt for pairwise classification.

```
{
  "role": "system",
  "content": "
    <INSTRUCTIONS>

    Examples:
    #########
    ---
    Text for Analysis:
    NAIROBI, Kenya (AP) _
    Suspected bombs [EVENT1]exploded[/EVENT1] outside the U.S. embassies in the
        Kenyan and Tanzanian capitals Friday, [EVENT2]killing[/EVENT2] dozens of
        people, witnesses said.
    --> before
    ---
    Text for Analysis:
    Suspected bombs exploded outside the U.S. embassies in the Kenyan and Tanzanian
        capitals Friday, killing dozens of people, witnesses [EVENT3]said[/EVENT3].
    The American ambassador to Kenya was among hundreds [EVENT12]injured[/EVENT12],
        a local TV said.
    --> after
    #########"
},
{
  "role": "user",
  "content": "
    ---
    Text for Analysis:
    <TEXT>
    ---

in one word -->"
}
```

Figure 4: Few-shot prompt for pairwise classification. <INSTRUCTIONS> is a placeholder for the instructions provided in Figure 3.

```
{
  "role": "system",
  "content": "
    Task Overview:
    You are given a text, in which some verbs are uniquely marked by [EVENT#ID]event
       [/EVENT#ID] (e.g., [EVENT1]event1[/EVENT1], [EVENT2]event2[/EVENT2]).
    Your task is to say which of the verbs happened first in a chronological order.
    More specifically, you need to return for each pair of verbs, which is two
       sentence apart,
    a single label out of the listed potential labels:
    before - the first verb happened before the second.
    after - the first verb happened after the second.
    equal - both verbs happened together.
    vague - It is impossible to know based on the context provided

    All responses should be valid and compact dot graph format.

    compact meaning:
    - do not mention transitive dependencies - if ei1 BEFORE ei2 and ei2 BEFORE ei3
       don't write ei1 BEFORE ei3
    - do not mention symmetric relation - if ei1 BEFORE ei2 don't write ei2 AFTER
       ei1"
},
{
  "role": "user",
  "content": "---
    Text for Analysis:
    The flu season is winding down, and it has [EVENT2]killed[/EVENT2] 105 children
       so far - about the average toll.
       The season [EVENT3]started[/EVENT3] about a month earlier than usual, [
          EVENT4]sparking[/EVENT4] concerns it might turn into the worst in a
          decade.
    ...
    ---

Respond only with valid dot graph format with the approprite markers and attributes
    (like label). Do not write an introduction or summary.
the graph:"
}
```

Figure 5: Zero-shot prompt for generating the entire temporal graph.

```
{
  "role": "system",
  "content": "
    <INSTRUCTIONS>

    Example:
    #########
    ---
    Text for Analysis:
    NAIROBI, Kenya (AP) _
    Suspected bombs [EVENT1]exploded[/EVENT1] outside the U.S. embassies in the
        Kenyan and Tanzanian capitals Friday, [EVENT2]killing[/EVENT2] dozens of
        people, witnesses [EVENT3]said[/EVENT3].
    ...
    ---
    the sample of correct labels are:

    digraph {
        "EVENT1" -> "EVENT2" [label="before"];
        "EVENT3" -> "EVENT12" [label="after"];
        "EVENT4" -> "EVENT5" [label="vague"];
        }
    #########"
},
{
  "role": "user",
  "content": "
    ---
    Text for Analysis:
    <TEXT>
    ---

Respond only with valid dot graph format with the approprite markers and attributes
    (like label). Do not write an introduction or summary.
the graph:"
}
```

Figure 6: Few-shot prompt for generating the entire temporal graph.

```
{
  "role": "system",
  "content": "
    Task Overview:
    You are given a text, in which some events are uniquely marked by [EVENT#ID]
        event[/EVENT#ID] (e.g., [EVENT1]event1[/EVENT1], [EVENT2]event2[/EVENT2]),
    and a dot graph which represent chronological order with error, where some edges
         form cycles.
    Your task is to decide which pair to drop (by his unique_id), being concise and
        removing the minimum number of edges.
    Pay attention, I used classifier to choose the most fitted relation (label
        attribute in dot graph)
    and score which represent the confidence of the classifier.

    relation meaning:
    before - the first verb happened before the second.
    after - the first verb happened after the second.
    equal - both events happen simultaneously
    vague - temporal order cannot be determined from the context"
},
{
  "role": "user",
  "content": "
    ---
    Text for Analysis:
    Barack Obama would make a great stand-up comic, not because he's the funniest
        president ever but because he uses jokes the same way many of us comedians
        do: as a weapon.

        Traditionally, the (intentionally) funny lines by our presidents have had
            one thing in common: They were self-deprecating. Sure, some presidents
            have [EVENT5]used[/EVENT5] jokes to take jabs at their opponents, but
            not to the extent of Obama.

        During his tenure, he has increasingly [EVENT8]unleashed[/EVENT8] biting
            comedic barbs against his critics and political adversaries. These jokes
             are [EVENT1000]intended[/EVENT1000] to do more than simply entertain
            you. They have an agenda.

        Obama's humor is often delivered the way a comedian dealing with a heckler
            would do it. He tries to undermine his opponents with it and get the
            crowd -- in this case the public -- on his side. I can [EVENT20]assure[/
            EVENT20] you that having a crowd laugh at your critic/heckler is not
            only effective in dominating them, it's also very satisfying.

    digraph Chronology {
        "EVENT5" -> "EVENT8" [label="BEFORE", score=0.71996284, unique_id=0];
        "EVENT8" -> "EVENT1000" [label="BEFORE", score=0.8759634, unique_id=1];
        "EVENT5" -> "EVENT20" [label="AFTER", score=0.9743732, unique_id=2];
        "EVENT1000" -> "EVENT20" [label="BEFORE", score=0.75076234, unique_id=3];
    }
    ---
Respond only with the unique_id list to drop (wrong label)"
}
```

Figure 7: Zero-shot prompt for cycle breaking.