Beyond Pairwise: Global Zero-shot Temporal Graph Generation

Alon Eirew[1] **Kfir Bar**[2] **Ido Dagan**[1]
[1]Computer Science Department, Bar-Ilan University
[2]Efi Arazi School of Computer Science, Reichman University
alon.eirew@gmail.com, kfir.bar@runi.ac.il

Abstract

Temporal relation extraction (TRE) is a fundamental task in natural language processing (NLP) that involves identifying the temporal relationships between events in a document. Despite the advances in large language models (LLMs), their application to TRE remains limited. Most existing approaches rely on *pairwise* classification, in which event pairs are considered individually, leading to computational inefficiency and a lack of global consistency in the resulting temporal graph. In this work, we propose a novel *zero-shot* method for TRE that generates a document's complete temporal graph at once, then applies transitive constraints optimization to refine predictions and enforce temporal consistency across relations. Additionally, we introduce **OmniTemp**, a new dataset with complete annotations for all pairs of targeted events within a document. Through experiments and analyses, we demonstrate that our method significantly outperforms existing zero-shot approaches while achieving competitive performance with supervised models.

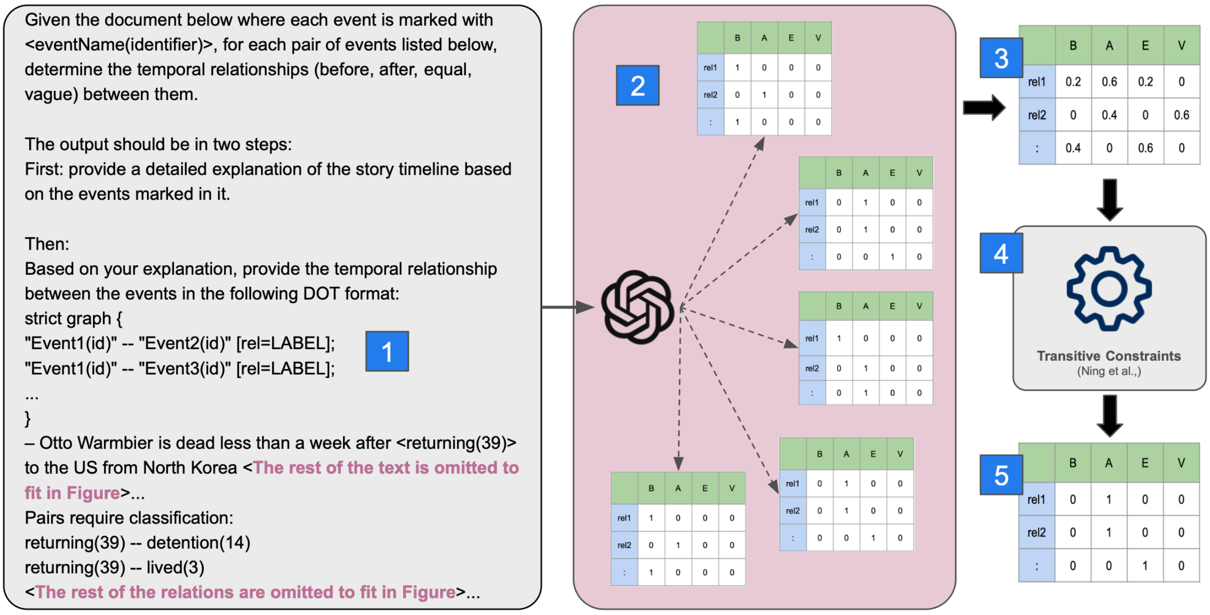
1 Introduction

Temporal relation extraction (TRE) is a fundamental task in natural language processing (NLP) that has been instrumental in various downstream tasks, including recent advancements in event forecasting (Ma et al., 2023), misinformation detection (Lei and Huang, 2023), and medical treatment timeline extraction (Yao et al., 2024).

TRE is formulated as follows: given a text with event mentions marked within it, identify all the temporal relations between these events. Accordingly, and ideally, a dataset for evaluating TRE models should consist of annotated relations between all pairs of events. However, annotating temporal relations is highly challenging (Pustejovsky and Stubbs, 2011), and *complete* annotation—where all possible event pairs in a document are labeled—has traditionally been considered unfeasible for human annotators (Naik et al., 2019). To manage this complexity, most datasets include labels for only a subset of event pairs, applying filtering methodologies such as restricting annotations to events within consecutive sentences (Chambers et al., 2014; Ning et al., 2018b) or creating temporal relation annotations through automated processes (Naik et al., 2019; Alsayyahi and Batista-Navarro, 2023). However, such restrictions can lead to unreliable model assessments, failing to accurately reflect a model's ability to capture long-range relations, or reinforce biases introduced by automated annotation techniques. Furthermore, incomplete annotations and the lack of global coverage have led the field to primarily focus on developing *pairwise* methods (Wen and Ji, 2021; Zhou et al., 2022), where a model extracts temporal relations between a single event pair at a time. Yet, such methods overlook the document's global temporal structure, resulting in inconsistencies in the output temporal graph (Wang et al., 2020), and are computationally inefficient, requiring $O(n^2)$ inference requests to predict all temporal relations across n given events.

Despite these challenges, TRE has seen significant progress in the development of supervised models (Tan et al., 2023; Niu et al., 2024). However, current utilization of LLMs remains limited, particularly in zero-shot settings (Kojima et al., 2022). The only existing studies (Yuan et al., 2023; Chan et al., 2024) have employed local pairwise prompting strategies, resulting in suboptimal results while also being time- and cost-inefficient. Consequently, the application of LLMs to TRE has been widely regarded as ineffective (Wei et al., 2024; Niu et al., 2024; Chan et al., 2024).

In response, we make the following two contributions. First, we demonstrate how to move beyond pairwise approaches by using LLMs. We introduce a novel zero-shot method that generates the entire



Given the document below where each event is marked with <eventName(identifier)>, for each pair of events listed below, determine the temporal relationships (before, after, equal, vague) between them.

The output should be in two steps:
First: provide a detailed explanation of the story timeline based on the events marked in it.

Then:
Based on your explanation, provide the temporal relationship between the events in the following DOT format:

```
strict graph {
"Event1(id)" -- "Event2(id)" [rel=LABEL];
"Event1(id)" -- "Event3(id)" [rel=LABEL];
...
}
```

– Otto Warmbier is dead less than a week after <returning(39)> to the US from North Korea <The rest of the text is omitted to fit in Figure>...
Pairs require classification:
returning(39) -- detention(14)
returning(39) -- lived(3)
<The rest of the relations are omitted to fit in Figure>...

	B	A	E	V
rel1	1	0	0	0
rel2	0	1	0	0
:	1	0	0	0

	B	A	E	V
rel1	0	1	0	0
rel2	0	1	0	0
:	0	0	1	0

	B	A	E	V
rel1	1	0	0	0
rel2	0	1	0	0
:	0	1	0	0

	B	A	E	V
rel1	1	0	0	0
rel2	0	1	0	0
:	1	0	0	0

	B	A	E	V
rel1	0	1	0	0
rel2	0	1	0	0
:	0	0	1	0

	B	A	E	V
rel1	0.2	0.6	0.2	0
rel2	0	0.4	0	0.6
:	0.4	0	0.6	0

	B	A	E	V
rel1	0	1	0	0
rel2	0	1	0	0
:	0	0	1	0

Figure 1: Illustration of the pipeline approach (§4): **[1]** We send the same prompt to GPT-4o to generate five separate instances of the document's *complete* temporal graph. **[2]** We extract the relation distribution as one-hot vectors over the temporal classes for each relation in each generation. **[3]** We sum and normalize the predictions into a single vector representing the joint prediction over the document's temporal graph. **[4]** We apply a transitive closure optimization algorithm to this vector. **[5]** The final temporal graph is obtained.

temporal graph *globally* in a single step. We then extend this basic zero-shot approach in two major ways. (1) We prompt the model to "think" by asking it to summarize the timeline of the given events in free-form language before generating the requested temporal classification labels for all event pairs. (2) We collect label distributions by running the model multiple times and then apply a global constraints algorithm that considers these distributions to produce a final globally optimal graph of relations (illustrated in Figure 1 and explained in §4). We show that our method significantly outperforms the existing zero-shot pairwise approach across most datasets while being more efficient, as shown in §6. Our findings demonstrate that, contrary to previous research, LLMs in zero-shot settings may be a valid alternative to supervised models for temporal relation extraction.

To address the incompleteness of temporal relation datasets, our second contribution is *OmniTemp*,[1] a new dataset that incorporates temporal relations between all pairs of targeted events to support unbiased evaluation (§3). Using this dataset, we provide an analysis that further highlights the importance of complete pairwise annotation for the reliable evaluation of temporal relation graph generation (§6.1).

2 Background

This section provides relevant background on datasets and zero-shot methods for the temporal relation extraction task.

2.1 Temporal Relation Extraction Datasets

The temporal relation extraction task aims to determine the temporal order between pre-extracted events in a text (Pustejovsky et al., 2003). For fair and unbiased model evaluation, datasets should provide gold labels for all event pairs or, at a minimum, be randomly sampled from the full set. However, most existing datasets for temporal relation extraction provide only partial annotation due to the complexity and cost of the process. As a result, the two most widely used datasets, MATRES (Ning et al., 2018b) and TimeBank-Dense (TB-Dense) (Chambers et al., 2014), annotate only relations between events in consecutive sentences.

Recently, the NarrativeTime project (Rogers et al., 2024) released a comprehensive, expert re-annotation of the TB-Dense corpus, covering all possible event pairs. The dataset includes seven relation types: *before*, *after*, *includes*, *is-included*, *equal*, *overlap*, and *vague*. Temporal relations are established based on event start times, end times, and durations. Notably, the *vague* relation indicates that the temporal relation cannot be determined

[1] The OmniTemp dataset will be made publicly available.

	Train	Test	All
Documents	20	10	30
Events	319	151	470
before	1,119	419	1,538
after	916	431	1,347
equal	90	60	150
vague	276	172	448
Total Relations	2,401	1,082	3,483

Table 1: OmniTemp dataset statistics.

from the provided context or where annotators disagree, and it is crucial for *complete* annotation, as it confirms that the pair was considered during annotation.

While NarrativeTime provides an exhaustively annotated dataset, it follows a complex annotation guidelines similar to TB-Dense. MATRES refined these guidelines by considering only event start times and reducing the label set to *before*, *after*, *equal*, and *vague*, improving inter-annotator agreement while offering an alternative and appealing setting for the task. However, MATRES is not exhaustively annotated. To bridge this gap, we develop OmniTemp, a dataset that follows the refined MATRES scheme while ensuring complete coverage of *all* event pairs across entire texts. Further details are provided in §3.

2.2 Zero-Shot Methods

Recent advancements in LLMs offer an opportunity to leverage their vast knowledge for zero-shot approaches (Kojima et al., 2022), enabling solutions without training data (Zhao et al., 2023). However, few studies have explored LLMs for temporal relation extraction in zero-shot settings. The most notable one is by Yuan et al. (2023), who applied a simple zero-shot chain-of-thought (ZS-CoT) method, where the model is asked about each relation for a given pair until it answers "yes". Another effort by Chan et al. (2024) experimented with prompt engineering and in-context learning. Both methods employed a pairwise approach and achieved suboptimal results on the MATRES and TB-Dense datasets. Additionally, the pairwise approach makes these methods cost- and time-inefficient.

One of our goals in this work is to provide a more efficient and effective alternative to pairwise approaches by processing the entire document globally in a single step (see §4).

3 The OmniTemp Dataset

OmniTemp is built following the MATRES (Ning et al., 2018b) approach; however, instead of annotating events only in consecutive sentences, the annotation is *complete*, covering all event pairs across the entire document. OmniTemp consists of a set of 30 English news summaries, written by humans (`Newser.com`), derived from the Multi-News dataset (Fabbri et al., 2019). We select summaries that portray large events, as these are rich in meaningful event mentions. Each summary contains a set of event mentions, with every pair assigned one of the following relations: *before*, *after*, *equal*, or *vague*. We now describe OmniTemp's annotation process (§3.1) along with dataset statistics (§3.2).

3.1 Annotation Process

For the annotation process, we hired three annotators, all non-expert native English speakers and either undergraduate or graduate students. We instruct annotators to follow the MATRES annotation guidelines, considering only "actual" events (e.g., *they won the game*). Events that are "non-actual", such as intentional, negated, recurring, conditional, or wishful (e.g., *I wish they win the game*), are excluded from annotation. Additionally, only the starting time of events is considered when establishing temporal relations.

The actual annotation was done on 30 news summaries, each containing approximately 500 words. The annotators used the EventFull annotation tool (Eirew et al., 2024), with all events in each document already highlighted. These events were extracted using the event detection method proposed by Cattan et al. (2021), which identifies all types of events (actual and non-actual) and extracts an average of 60 event mentions per document, forming the initial set of events. We follow the same annotation protocol as proposed in EventFull.[2] First, the annotation process begins with the selection of 15 to 18 of the most salient "actual" events from each story.[3] After selecting these events, each document was annotated for temporal relations (*before*, *after*, *equal*, or *vague*) by all three annotators. Finally,

[2] The complete annotation guidelines are available within the EventFull annotation tool.

[3] Eirew et al. (2024) found that beyond 18 events, annotation becomes challenging for non-expert annotators. This event reduction aligns with previous efforts to decrease annotation workload by limiting the number of events considered (Chambers et al., 2014; Ning et al., 2018b; Tan et al., 2024).

majority voting was used to determine the final relation, and in cases of disagreement, the relation was labeled as *vague*. Further details about the annotators, time and cost are provided in Appendix D.

3.2 Dataset Statistics and Comparison

Table 1 summarizes the OmniTemp dataset's statistics. Overall, the final annotated version of OmniTemp consists of 30 documents, corresponding to 470 event mentions and 3,483 relations. Appendix I, Table 6 presents the statistics of prominent datasets for the temporal relation extraction task alongside OmniTemp.

The agreement among our annotators averaged 0.72 kappa (Fleiss and Cohen, 1973), corresponding to substantial agreement and is comparable to that of TB-Dense (Chambers et al., 2014) (0.56κ–0.64κ), NarrativeTime (Rogers et al., 2024) (0.68κ), TDD-Manual (Naik et al., 2019) (0.69κ), and MATRES (Ning et al., 2018b) (0.84κ). Additionally, to verify annotation accuracy, one of the authors re-annotated 50 random pairs, with 46 matching the majority vote of the annotators, further confirming the high quality of the annotations.

Finally, as mentioned, our motivation for developing OmniTemp was to provide complete annotations within each document, similar to NarrativeTime (§2.1). However, in datasets such as MATRES and TB-Dense, where annotation is complete only between consecutive sentences, event pairs may be inferred through transitivity rules. The extent to which this automatic inference scales, however, remains unclear. To investigate this, we analyze the NarrativeTime dataset by considering all relations within the same sentence or between consecutive sentences. We then apply a transitive algorithm to infer additional relations and assess how many can be recovered beyond a single sentence. Our analysis shows that while some long-distance relations are recovered, most inferred relations remain within close proximity and occur infrequently. This finding highlights the importance of exhaustive annotation in constructing more complete and accurate story timelines. The full analysis is provided in Appendix E.

4 Zero-Shot Temporal Graph Generation

ZSL-Global. Our approach begins with a simple yet ambitious idea: prompting an LLM[4] to generate the full temporal graph of a document in a single call. This initial method, which we call ZSL-Global (prompt is provided in Appendix I), follows a straightforward zero-shot approach where the model is explicitly instructed to produce relations for all event pairs at once.

To implement this, we structure the prompt, following Yuan et al. (2023). It begins with a general instruction about the task, explaining that the model needs to extract temporal relations between events. The entire document is then provided, with the relevant event mentions highlighted using angle brackets assigned unique identifiers (e.g., '<attack(7)>'). Finally, to create a realistic scenario where relations between all event pairs are desired, we include all possible pairs instead of only the gold-labeled ones. When the number of event pairs exceeds 100–200, depending on the number of events in the document, we divide them into groups of up to 200, to fit the output length constraints of the LLM. Each group is processed in a separate call to the LLM, with the full document provided alongside the subset of event pairs. Further details on how event pairs are segmented are provided in Appendix A. For the output, we instruct the model to represent relations as a graph, where events serve as nodes and relations as links, and format it in the DOT language (Gansner, 2006) to facilitate parsing.

ZSL-Timeline. While the previous method provides a strong baseline, the model sometimes produces incorrect relations even when clear temporal cues exist in the text. To improve accuracy, rather than directly generating a structured graph, we first ask the model to construct a free-form timeline—an unstructured summary describing the sequence of only the marked events in natural language (illustrated in Figure 1). This approach is inspired by reasoning-based prompting techniques (Wang et al., 2023a; Sun et al., 2024). By generating the timeline first, the model gains a broader understanding of the temporal flow before making explicit classification decisions. Once the timeline is generated, the model then starts generating the temporal relations for all event pairs in the DOT format, as before. This process encourages the model to "think" before assigning relations. We call this method ZSL-Timeline (an example of the generated timeline is presented in Appendix I, Figure 7).

ZSL-SelfConsistency. LLMs are inherently stochastic and may generate different labels for the same input when run multiple times. This variability can lead to unstable outputs, particularly for

[4] In this work we experiment with GPT-4o https://platform.openai.com.

event pairs that are naturally difficult to classify. To address this, we incorporate self-consistency prompting (Wang et al., 2023b), where we run the model five[5] times on each input and aggregate the results using majority voting to determine the most frequently classified relation for each event pair. This method, which we call ZSL-SelfConsistency, improves robustness by reducing randomness.

ZSL-GlobalConsistency. While self-consistency helps stabilize the model's predictions, it does so by focusing only on individual label distributions. Although the LLM considers global context when predicting all relations at once, majority voting treats each relation independently, disregarding dependencies between temporal relations. To address this, we replace majority voting with transitive constraints using an Integer Linear Programming optimization algorithm (Ning et al., 2018a), which enforces global consistency by resolving conflicts (For example, if event *A* precedes event *B* and *B* precedes event *C*, then *A* must also precede *C*), while optimizing overall likelihood over the predicted classifications. By applying these rules, the algorithm ensures that the final temporal graph satisfies transitivity and maintains logical consistency across all event pairs. A formal description of this process is provided in Appendix F.

Through this evolution—from a basic zero-shot approach to a structured, globally consistent method—we develop an increasingly effective strategy for leveraging LLMs in generating global temporal relation graphs. In §6, we evaluate each method individually for both accuracy and consistency.

5 Experimental Setting

We describe the datasets and models used in our experiments. Technical details are in Appendix A.

5.1 Datasets

In our experiments, we use our own OmniTemp and three additional datasets: MATRES, TB-Dense, and NarrativeTime. Notably, TCR (Ning et al., 2018a) and TDD-Manual (Naik et al., 2019), two additional datasets for the TRE task, are excluded from our experiments as they omit the *vague* relation. Since we generate relations for all possible event pairs, the *vague* label is essential to avoid forcing incorrect relations when context is insufficient (further details on these datasets are presented in Appendix I). Below, we provide details on the datasets used in our experiments. For our own OmniTemp, we use the first 10 documents as the test set and the remaining documents as the training set, while for all other datasets, we follow their predefined splits.

MATRES. In MATRES, only events within consecutive sentences are annotated. The dataset includes four relation types: *before*, *after*, *equal*, and *vague*, with temporal relations determined based on event start times.

TB-Dense. Similar to MATRES, only events within consecutive sentences are annotated in the TB-Dense dataset. It includes six relation types, the four from MATRES plus *includes* and *is-included*. Temporal relations are determined based on event start and end times as well as their duration.

NT-6. The NarrativeTime (NT) dataset, previously introduced in §2.1, features seven relation types, including the six from TB-Dense and the *overlap* relation. However, we exclude the *overlap* relation as it is incompatible for the transitive consistency methods, given that the symmetric counterpart was not annotated. Additionally, NT documents contain an average of 50 events, corresponding to 1,200 relations, per document. This poses challenges for LLMs due to context length limitations. To address this, we randomly select only 18 events per document. Further details on these decisions are provided in Appendix H. We refer to this dataset as NT-6 as it retains only six relations.

5.2 Baseline and State-of-the-Art Models

We compare our zero-shot methods with four models, reproducing state-of-the-art (SOTA) supervised models and a zero-shot chain-of-thought (ZS-CoT) baseline method.

Bayesian (Tan et al., 2023). Bayesian-Translation is the current publicly available state-of-the-art pairwise model for temporal relation extraction. It leverages a COMET-BART encoder (Hwang et al., 2020) and a graph translation model (Balazevic et al., 2019) to incorporate prior knowledge from the ATOMIC commonsense knowledge base, refining event representations for relational embedding learning. Additionally, it employs a Bayesian framework to estimate the uncertainty of the learned relations.

RoBERTa (Tan et al., 2023). A strong pairwise model for temporal relation extraction, similar in

[5] We generate five outputs based on experimental results showing that performance saturates beyond this point.

Model	MATRES	TB-Dense	NT-6	OmniTemp
Supervised SOTA Pairwise Models				
RoBERTa (Tan et al.)	78.9	**56.9**	59.3	73.6
Bayesian (Tan et al.)	**80.6**	55.2	64.9	78.7
Bayesian + Constraints	79.2	55.8	**65.6**	**80.7**
Zero-Shot Prompting with GPT-4o				
CoT (Yuan et al.)	56.6	**42.8**	49.3	67.2
ZSL-Global (Ours)	59.0	37.7	48.4	62.3
ZSL-Timeline (Ours)	58.4	39.1	52.2	68.5
ZSL-SelfConsistency	58.0	39.3	55.6	72.4
ZSL-GlobalConsistency	**63.0**	**42.8**	**58.4**	**74.5**

Table 2: F1 scores of all models on the four datasets. We use the F1 definition of (Ning et al., 2019). Further details on experiment results are in Appendix B.

	Inconsistencies	Time	Cost
CoT (Yuan et al.)	29	420	0.70
ZSL-Global	10	60	0.03
ZSL-Timeline	7	70	0.03
ZSL-SelfConsistency	9	350	0.15
ZSL-GlobalConsistency	0	354	0.15

Table 3: The average time (seconds), cost ($), and number of transitive inconsistencies when applying different methods to generate a temporal graph from a document in the OmniTemp dataset.

architecture to the Bayesian model described above, but replacing the COMET-BART encoder with a RoBERTa-large encoder (Zhuang et al., 2021). Unlike the Bayesian model, it learns relational embeddings without relying on prior knowledge from external sources. We use this model as it represents a strong, purely supervised approach, allowing for a direct comparison without the influence of external knowledge.

Bayesian + Constraints. We extend the Bayesian model with the transitive constraints optimization algorithm (Ning et al., 2018a), the same algorithm used in our ZSL-GlobalConsistency method, applying it at inference time to enable a more direct comparison with our self and global consistency methods.

CoT (Yuan et al., 2023). As a baseline model, we re-implemented the CoT model (Yuan et al., 2023) using GPT-4o, replacing the original implementation, which used ChatGPT. To the best of our knowledge, this is the strongest zero-shot approach for temporal relation extraction.

For evaluation, we report the F1 score on all datasets following the definition in (Ning et al., 2019), where the *vague* relation is excluded from true positive predictions.

6 Results

Our results are presented in Table 2, with *supervised* SOTA pairwise models in the upper section and our *zero-shot* GPT-4o results in the lower section. Overall, our ZSL-GlobalConsistency approach (§4) outperforms the CoT baseline (Yuan et al., 2023) by a large margin across all datasets except TB-Dense (see §6.1 for further analysis of TB-Dense). On dense datasets, NT-6 and OmniTemp, ZSL-GlobalConsistency achieves competitive performance compared to the supervised RoBERTa model (74.5 vs. 73.6 for OmniTemp and 58.4 vs. 59.3 for NT-6), while requiring no training data. The supervised SOTA Bayesian model performs better than our approach on these datasets, but notably it depends not only on training data but also on a substantial external common-sense knowledge base, which may not be applicable for many domains and languages. This positions ZSL-GlobalConsistency as an appealing zero-shot alternative for temporal relation extraction in domains lacking labeled training data or comprehensive knowledge bases.

Table 3 demonstrates, over the OmniTemp dataset, the effectiveness of our approach in terms of time, cost, and temporal consistency of the generated graphs, comparing it to the prior zero-shot CoT baseline. To assess effectiveness, we evaluate each method by measuring the average per document for: (1) generation time, (2) cost, calculated using OpenAI's billing system, and (3) transitive consistency. The latter is evaluated by applying a transitive closure algorithm (Warshall, 1962) and counting transitive contradictions—relations that violate the transitivity constraints defined by Ning et al. (2018a). The results show that all the evolving versions of our method are more cost- and time-efficient than the baseline CoT method. Moreover, the temporal graphs generated by all our methods are significantly more consistent than those produced by the baseline, reinforcing that prompting the LLM to generate the full graph in one step enables more effective use of global information. Each refinement of our method—from a simple zero-shot prompting approach to our final version, ZSL-GlobalConsistency—leads to improvements in both consistency and accuracy, which are key aspects of temporal relation extraction, while also increasing efficiency compared to the baseline.

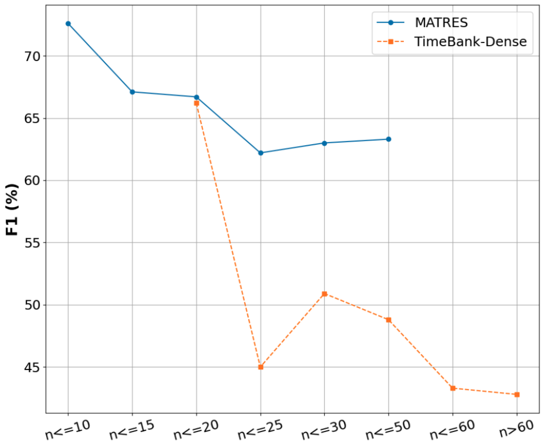
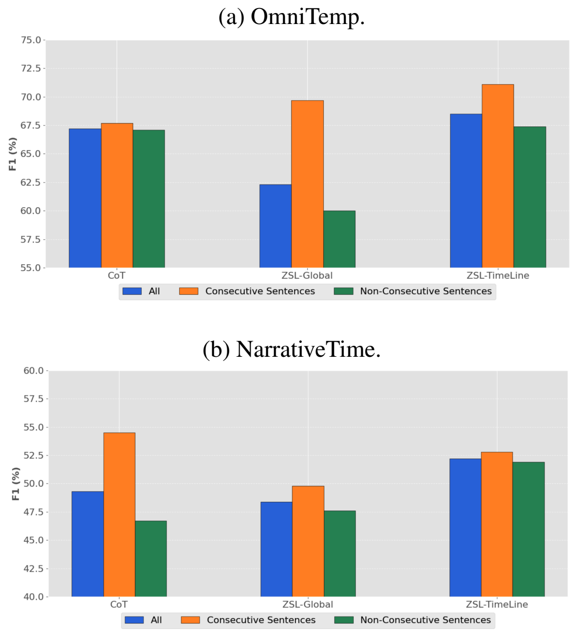


Figure 2: Impact of event count per document on ZSL-GlobalConsistency performance, evaluated on MATRES and TB-Dense. The x-axis is cumulative, and the y-axis shows the F1 score per subset.

6.1 Qualitative Analysis

Event Mentions Count. We examine how the number of events in a document affects the performance of our ZSL-Timeline method. Our hypothesis is that models encoding global information are influenced by the number of events, as they process more information at once. In contrast, pairwise methods, which consider one event pair at a time, are likely less affected. In Figure 2, we group MATRES and TB-Dense documents into subsets of increasing event counts.[6] Overall, performance declines as the number of events increases, supporting our hypothesis. However, in the TB-Dense curve, there is a sharper drop for documents with more than 25 events, deviating from the trend. A closer look reveals that these additional TB-Dense documents contain mostly *vague* relations. Further analysis, summarized in Figure 4, indicates that ZSL-Timeline struggles with *vague* relations, particularly in the TB-Dense and MATRES corpora. Notably, the *vague* relation is particularly challenging and often associated with annotator disagreement (Chambers et al., 2014).

Event Pair Distance. From Table 2 we learn that our initial version ZSL-Global outperforms the CoT baseline on MATRES but underperforms on OmniTemp. This difference may arise from the two annotation styles of MATRES and OmniTemp, with the former restricting the distance between events to at most *one* sentence, while the latter imposes no such limitation. To explore this, we evaluate CoT, ZSL-Global, and ZSL-Timeline on three subsets of OmniTemp and NT-6: the full dataset, only event pairs where the distance between them is at most one sentence (consecutive-sentences), and only event pairs where the distance between them is greater than one sentence (non-consecutive-sentences). See Figure 3.[7]


(a) OmniTemp.

(b) NarrativeTime.

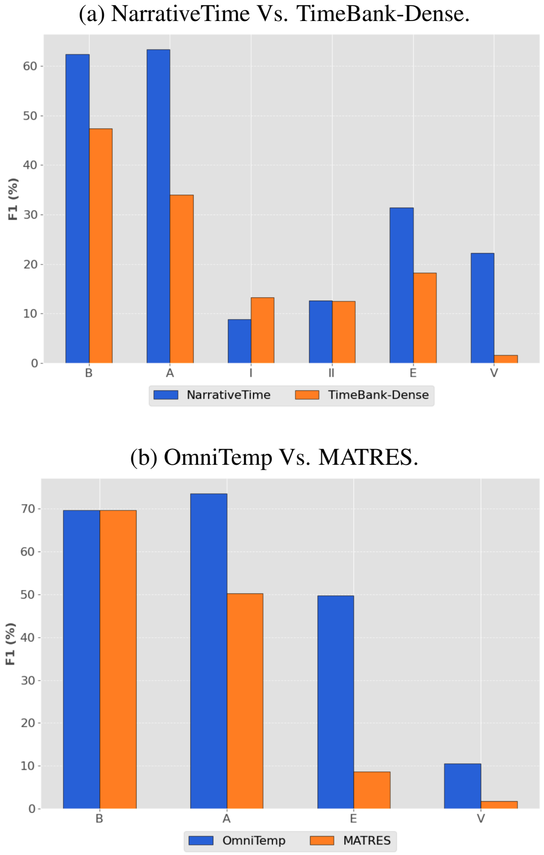
Figure 3: Performance across different relation sets: (1) consecutive-sentence events, (2) non-consecutive-sentence events, and (3) full-document relations.

Our findings show that on the four-relation OmniTemp dataset, the CoT baseline performs consistently across all sentence distances, while ZSL-Global performs significantly better on consecutive-sentence relations, with a 10-point gap compared to non-consecutive ones. This discrepancy explains why ZSL-Global improves performance on MATRES, which is annotated within consecutive sentences, but underperforms on OmniTemp, which spans entire documents (Table 2).

Interestingly, the opposite trend occurs in the more challenging six-relation NT-6 dataset, where the CoT baseline performs much better on consecutive-sentence relations, with a 8-point gap compared to performance on the non-consecutive-sentences subset. Noteworthy, in both cases ZSL-Timeline helps mitigating these issues—especially in NT-6—leading to overall improvements across entire documents. We stress that these findings

[6] The other datasets we experimented with, have a limited number of events per instance.

[7] For a fair comparison, we compare CoT to our methods before applying global consistency, isolating its performance from that achieved through transitive constraints, whose effectiveness depends on the quality of the input relations.



(a) NarrativeTime Vs. TimeBank-Dense.

(b) OmniTemp Vs. MATRES.

Figure 4: ZSL-Timeline performance per relation type across two datasets with similar annotation schemes. Six-label datasets (TB-Dense and NT-6) and four-label datasets (MATRES and OmniTemp). The relations are denoted as: A = *after*, B = *before*, I = *includes*, II = *is-included*, E = *equal*, and V = *vague*.

highlight the need for document-level annotations for reliable evaluation of temporal relation classification, especially in zero-shot settings where models cannot rely (not realistically) on distribution patterns in the annotations.

Label Inconsistency. The performance gap between our methods and the supervised models varies across datasets, being more pronounced in MATRES and TB-Dense than in NT-6 and OmniTemp. To better understand this gap, we analyze the ZSL-Timeline performance per label, grouping datasets with similar label categories and comparing them, as shown in Figure 4. Our ZSL-Timeline method performs significantly worse on MATRES and TB-Dense than on OmniTemp and NT-6.

To investigate this further, we examine label consistency in documents and event pairs shared between TB-Dense and MATRES, which annotated the same corpus. There are 983 such event pairs. While these datasets follow different annotation guidelines, certain labels should remain consistent. For instance, if an event pair is labeled *equal* in TB-Dense—indicating that both the start and end times of the two events are the same—then the relation should also be *equal* in MATRES. Measuring consistency across the four shared relations, we find strong agreement for *before* and *after*, with *before* being the most consistently annotated. However, significant inconsistencies were evident in *vague* and *equal*. Detailed results are provided in Appendix C. Since in zero-shot settings the model is not trained on a dataset, it does not learn dataset-specific biases. The annotation inconsistency between MATRES and TB-Dense may partly explain the performance drop on these datasets, particularly for *vague* and *equal* relations, as well as the lower performance on *after* compared to *before*. This analysis, along with the pair distance analysis, raises a broader question of whether the evaluation of zero-shot approaches on TB-Dense and MATRES is sufficiently reliable.

In conclusion, our method outperforms the previous zero-shot approach, especially on the two more reliable datasets with complete and consistent annotations.

7 Conclusion

In this work, we introduced a novel zero-shot LLM approach for temporal relation extraction that generates the entire temporal graph at once. Our method moves beyond traditional pairwise approaches, which suffer from computational inefficiency and lack a global perspective. To ensure temporal consistency in predictions, we incorporated self-consistency prompting and transitive constraints optimization, significantly improving both accuracy and efficiency while generating relations completely free of inconsistencies. Our results show that zero-shot LLMs, when prompted to generate the timeline of events in free-form language before assigning labels to event pairs and extended with a global constraints algorithm, can serve as a viable alternative to supervised models, particularly in domains without annotated data. Additionally, we introduced *OmniTemp*, a new dataset with complete annotations for all event pairs, following the refined annotation guidelines of MATRES. By providing gold labels for every event pair in a document, this dataset enables a fair evaluation of zero-shot approaches.

Limitations

While our proposed zero-shot temporal graph generation approach demonstrates significant advantages over pairwise methods, several limitations remain that warrant further investigation.

First, closed LLMs such as GPT-4o do not disclose their training data. Therefore, results on the three datasets we investigate may be affected by potential data contamination if their test sets were included in GPT's training phase. However, OmniTemp is a completely new resource that is not yet publicly available, ensuring uncontaminated results.

Second, although self-consistency prompting mitigates stochasticity to some extent, the model's responses can still be inconsistent, especially when handling long-distance temporal dependencies or ambiguous event relations.

Third, the computational cost of using LLMs for large-scale inference remains a challenge. While our approach significantly reduces costs compared to pairwise methods, generating a full temporal graph for documents with many events can still be time-intensive and expensive, particularly when applying self-consistency with multiple generations.

Fourth, in this research, we present our results on GPT-4o; however, we expect similar conclusions with other equivalent LLMs.

Finally, our dataset, **OmniTemp**, provides exhaustive event-event relation annotations following the MATRES-style four-relation schema (*before, after, equal, vague*), considering only the start time of events. As a result, it may not represent all TRE tasks, such as those requiring the *includes* relation or those that also consider event end times and durations.

Despite these limitations, our study highlights promising directions for leveraging LLMs in structured event reasoning and lays the groundwork for future improvements in temporal relation extraction.

References

James F. Allen. 1984. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154.

Sarah Alsayyahi and Riza Batista-Navarro. 2023. TIMELINE: Exhaustive annotation of temporal relations supporting the automatic ordering of events in news articles. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16336–16348, Singapore. Association for Computational Linguistics.

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019. Multi-relational poincaré graph embeddings. In *Neural Information Processing Systems*.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2021. Cross-document coreference resolution over predicted mentions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5100–5107, Online. Association for Computational Linguistics.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Chunkit Chan, Cheng Jiayang, Weiqi Wang, Yuxin Jiang, Tianqing Fang, Xin Liu, and Yangqiu Song. 2024. Exploring the potential of ChatGPT on sentence level relations: A focus on temporal, causal, and discourse relations. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 684–721, St. Julian's, Malta. Association for Computational Linguistics.

Alon Eirew, Eviatar Nachshoni, Aviv Slobodkin, and Ido Dagan. 2024. EventFull: Complete and consistent event relation annotation.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Joseph L. Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and Psychological Measurement*, 33(3):613–619.

Emden R. Gansner. 2006. Drawing graphs with dot.

Gurobi Optimization, LLC. 2024. Gurobi Optimizer Reference Manual.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2020. COMET-ATOMIC 2020: On symbolic and neural commonsense knowledge graphs. In *AAAI Conference on Artificial Intelligence*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Yuanyuan Lei and Ruihong Huang. 2023. Identifying conspiracy theories news based on event relation graph. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9811–9822, Singapore. Association for Computational Linguistics.

Yunshan Ma, Chenchen Ye, Zijian Wu, Xiang Wang, Yixin Cao, and Tat seng Chua. 2023. Context-aware event forecasting via graph disentanglement. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

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.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003. TimeML: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

James Pustejovsky and Amber Stubbs. 2011. Increasing informativeness in temporal annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 152–160, Portland, Oregon, 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.

Simeng Sun, Yang Liu, Shuohang Wang, Dan Iter, Chenguang Zhu, and Mohit Iyyer. 2024. PEARL: Prompting large language models to plan and execute actions over long documents. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 469–486, St. Julian's, Malta. Association for Computational Linguistics.

Xingwei Tan, Gabriele Pergola, and Yulan He. 2023. Event temporal relation extraction with Bayesian translational model. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1125–1138, Dubrovnik, Croatia. Association for Computational Linguistics.

Xingwei Tan, Yuxiang Zhou, Gabriele Pergola, and Yulan He. 2024. Set-aligning framework for autoregressive event temporal graph generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3872–3892, Mexico City, Mexico. Association for Computational Linguistics.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020. Joint constrained learning for event-event relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 696–706, Online. Association for Computational Linguistics.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for 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*.

Stephen Warshall. 1962. A theorem on boolean matrices. *J. ACM*, 9(1):11–12.

Kangda Wei, Aayush Gautam, and Ruihong Huang. 2024. Are LLMs good annotators for discourse-level event relation extraction? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1–19, Miami, Florida, USA. Association for Computational Linguistics.

Haoyang Wen and Heng Ji. 2021. Utilizing relative event time to enhance event-event temporal relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10431–10437, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiarui Yao, Harry Hochheiser, WonJin Yoon, Eli Goldner, and Guergana Savova. 2024. Overview of the 2024 shared task on chemotherapy treatment timeline extraction. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 557–569, Mexico City, Mexico. Association for Computational Linguistics.

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.

Xuandong Zhao, Siqi Ouyang, Zhiguo Yu, Ming Wu, and Lei Li. 2023. Pre-trained language models can be fully zero-shot learners. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15590–15606, Toronto, Canada. 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.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Experimental Details

For all supervised model experiments, we follow the experimental setup of Tan et al. (2023). To this end, we conducted a grid search to determine the optimal hyperparameters and embedding dimensionality for each test. Each training episode was run for 50 epochs on a single A100 GPU,[8] with the best-performing epoch on the development set selected for evaluation. For the GPT-4o experiments, we use 'gpt-4o-2024-08-06' version through OpenAI API. In all experiments, we provide the model with all event pairs combinations, and evaluate on the available gold labels. For the MATRES and TimeBank-Dense (TB-Dense) datasets, we evenly divide the set of pairs in documents containing more than 20 events. In TB-Dense, for documents exceeding 40 events, we further group the pairs into sets of 100. Finally, In cases the generation missed pairs or is malformed, we regenerate the document or its respective split. For transitive constraint optimization, we employ the Gurobi Optimizer (Gurobi Optimization, LLC, 2024).

Model	MATRES	TB-Dense	NT-6	OmniTemp
ZSL-Global (Ours)	59.0±1.4	37.7±1.8	48.4±2.5	62.3±0.5
ZSL-Timeline (Ours)	58.4±2.4	39.1±0.7	52.2±2.8	68.5±1.0

Table 4: F1 scores of ZSL-Global and ZSL-Timeline are reported along with the standard deviation.

	Train	Dev	Test
MATRES	13,577	NA	837
TB-Dense	4,205	649	1,451
NarrativeTime	68,317	2,759	7,925

Table 5: Statistics of event-event relations in the datasets used in this study.

B Further Details on Reported Results

We provide further details on the results presented in Table 2. For the supervised models—RoBERTa, Bayesian, and Bayesian + Constraints—we report the best results achieved following a hyperparameter search (further detailed in Appendix A). For the CoT experiment, we conducted a single evaluation run for each dataset and used this result. Constructing an ensemble or computing the mean for this experiment across multiple runs was beyond our budget. Additionally, our model results are substantially higher, making further aggregation unnecessary. In Table 4, we report the results for ZSL-Global and ZSL-Timeline, presenting the mean result obtained from five generations along with the standard deviation. For ZSL-SelfConsistency and ZSL-GlobalConsistency, we conducted a single run for each experiment, similar to CoT, as these experiments are more costly, and the observed standard deviation does not justify the additional expense.

C Label Inconsistency Evaluation

We describe the *Label Inconsistency* experiment detailed in §6.1. MATRES (Ning et al., 2018b) and TB-Dense (Chambers et al., 2014) annotate the same set of 35 documents but follow different an-

[8] Experiment GPU time varies depending on the size of the training set, ranging from 1 to 20 hours for a full training episode.

notation schemes. MATRES considers only event start times to determine temporal order, while TB-Dense accounts for event start times, end times, and durations.

To isolate this difference, we define the following ground truth for each relation: (1) If a pair is marked as *vague* in MATRES, meaning the event start time is unclear, the same pair should also be *vague* in TB-Dense since both the start time and duration are uncertain. (2) If a pair in TB-Dense is annotated as *before*, *after*, or *equal* based on both start and end times, the corresponding MATRES annotation should reflect the same relation when considering only event start times. Figure 5 presents our findings in terms of label consistency and inconsistency between the two datasets.

D Annotation Costs and Time

For the annotation process of OmniTemp (detailed in §3.1), we hired three student annotators (two males and one female) to label temporal relations between event pairs. Their location will be revealed once anonymity requirements are lifted. The total annotation time for OmniTemp, including onboarding, amounted to 85 hours, with each worker paid $15 per hour (which is considered a fair market value in their region).

E Filling Transitive Relations

As discussed in §3.2, to assess the coverage achievable by inferring transitive relations in resources annotated only with consecutive sentences, we extracted from NarrativeTime only the relations between event pairs in consecutive sentences. We then applied a transitive closure algorithm (Warshall, 1962) to construct additional relations and compared the results with the original set of relations. Figure 8 presents the experimental results.

F Formal Description of ZSL-GlobalConsistency

ZSL-GlobalConsistency is formulated as follows: we run the ZSL-Timeline method five times on each input as described in §4, generating five temporal graphs per document, denoted as $G = \{g_1, \ldots, g_5\}$ where each g_n represents a labeled directed graph parsed from the DOT-language output. Each graph consists of a set of predicted event-pair relations: $g_n = \{p_{12}, p_{13}, \ldots, p_{23}, p_{24}, \ldots, p_{nm}\}$ where each relation p_{ij} is represented as a one-hot vector over the six relation types. We then sum these vectors element-wise across all five graphs and normalize them to obtain a single distribution per event pair: $d_{ij} = \frac{1}{5}\sum_{n=1}^{5} p_{ij}^{(n)}$ where each d_{ij} represents the normalized label distribution for the event pair (e_i, e_j). Instead of selecting the most frequent relation via majority voting, we apply the transitive constraints optimization algorithm, which returns a temporally consistent graph. We call this final method ZSL-GlobalConsistency (Figure 1).

G Dataset Licenses and Sources

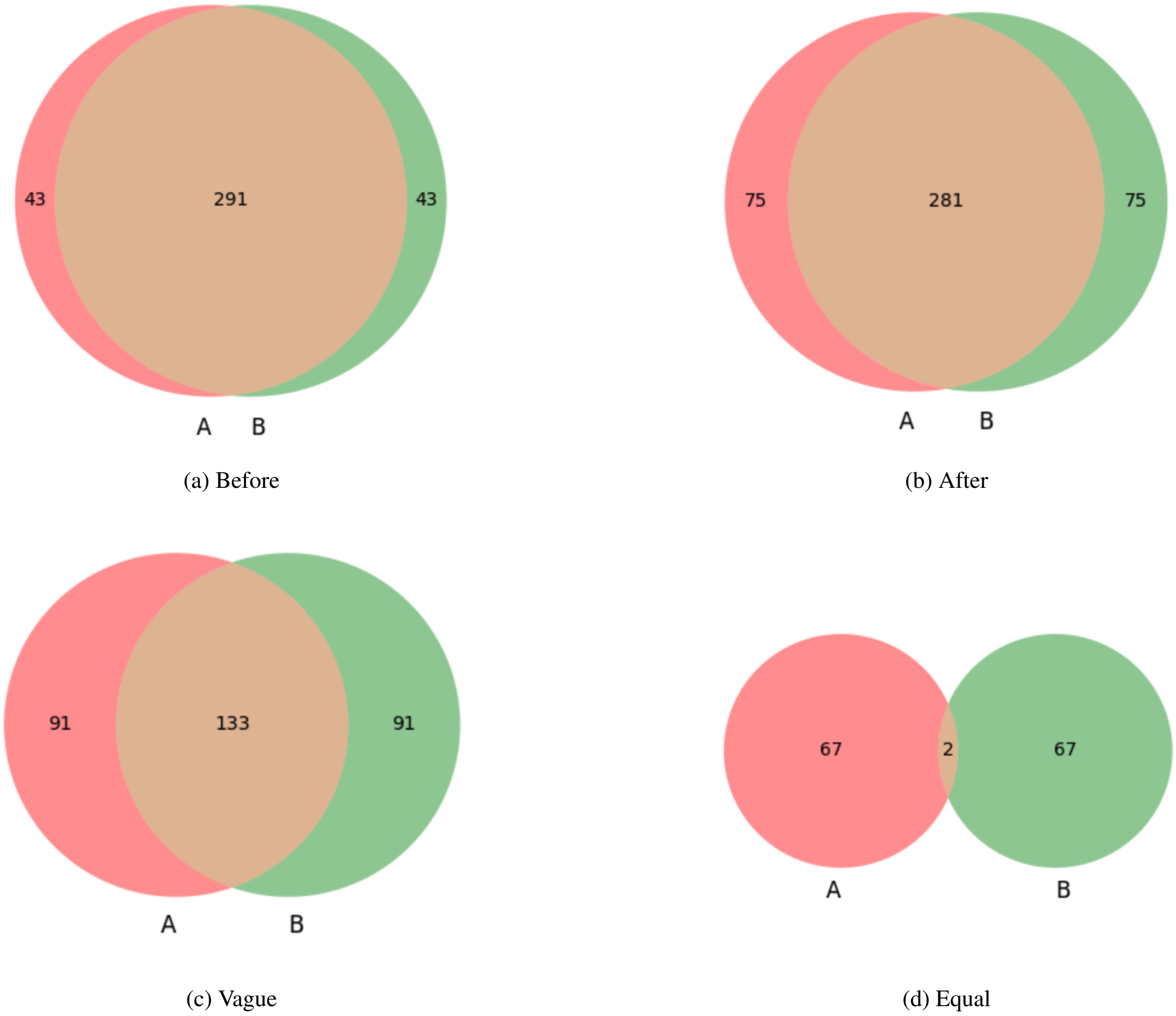
In our experiments, we use the following commonly used datasets for evaluating the temporal relation extraction task: MATRES (Ning et al., 2018b), provided without a license; TimeBank-Dense (Chambers et al., 2014), provided without a license; and NarrativeTime (Rogers et al., 2024), provided under the MIT license. Additionally, OmniTemp uses summaries from the Multi-News corpus (Fabbri et al., 2019), which is distributed under a custom license that permits free academic use. All datasets were downloaded from official repositories, and used appropriately. OmniTemp will also be released under a free-to-use academic license.

H Adjustments to the NarrativeTime Dataset

The NarrativeTime (NT) dataset, introduced in §2.1, features seven relation types, including the six from TB-Dense and the *overlap* relation. Our temporal consistency algorithm relies on Allen's transitivity laws (Allen, 1984), which require each relation type to have a symmetric counterpart (e.g., if event *A* occurs *before* event *B*, then *B* must occur *after* *A*). However, the *overlap* relation in NT lacks a symmetric counterpart, making it incompatible for transitive consistency methods. Therefore, before using NT, we exclude event pairs labeled with the *overlap* relation. Additionally, NT documents contain an average of 50 event mentions per document, corresponding to approximately 1,100 relations, which makes them difficult to process with LLMs due to context length limitations. Handling such documents requires segmenting them and making individual calls to the model for each segment, which increases costs, as discussed in §4. To avoid segmentation and reduce costs, we randomly select 18 events per document from the test set, along with all their associated relations. The choice of 18 events was based on empirical observations, as it represents the maximum number that can typically fit within the model's context window without requiring segmentation. This reduction is not applied to the training set, which we use to fine-tune the supervised models. We refer to this pre-processed version as NT-6, as it retains only six relation types.

I Additional Experiment Tables and Figures

Table 6 presents a comparison between common datasets used for evaluating models on the temporal relation task alongside OmniTemp. Table 5 presents the split statistics of these datasets. Figure 6 presents an example of the ZSL-Global prompt. Figure 7 presents an example of the generated timeline using the ZSL-Timeline approach.



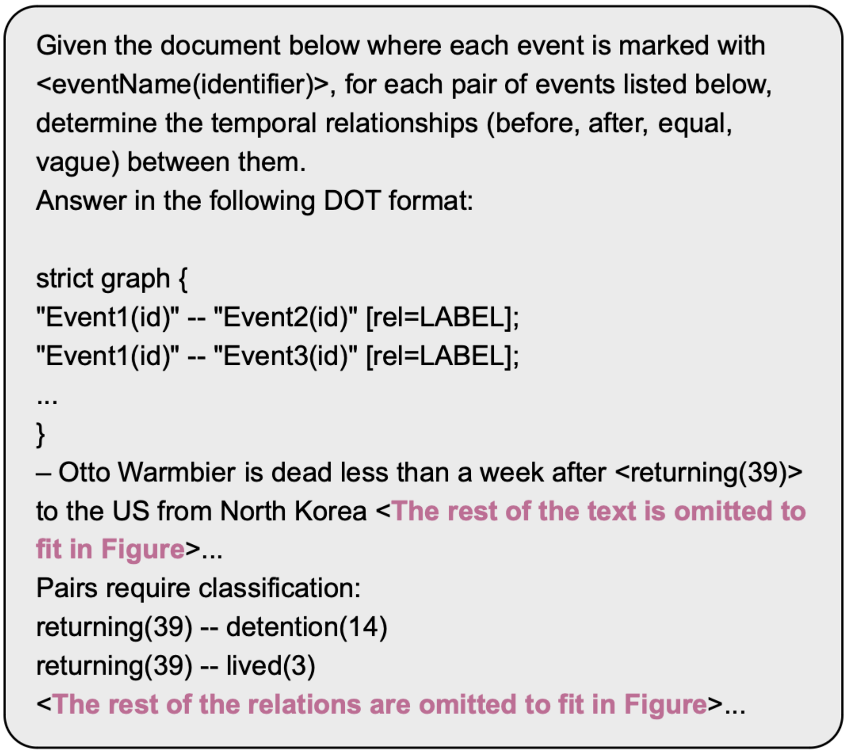
(a) Before

(b) After

(c) Vague

(d) Equal

Figure 5: Label Inconsistency: Each group, A and B, represents MATRES and TimeBank-Dense respectively. The intersecting area indicates consistency in label annotation between the two datasets, with the number of such pairs highlighted in the middle, while the non-intersecting areas represent pairs assigned different labels in each dataset.



Given the document below where each event is marked with <eventName(identifier)>, for each pair of events listed below, determine the temporal relationships (before, after, equal, vague) between them.
Answer in the following DOT format:

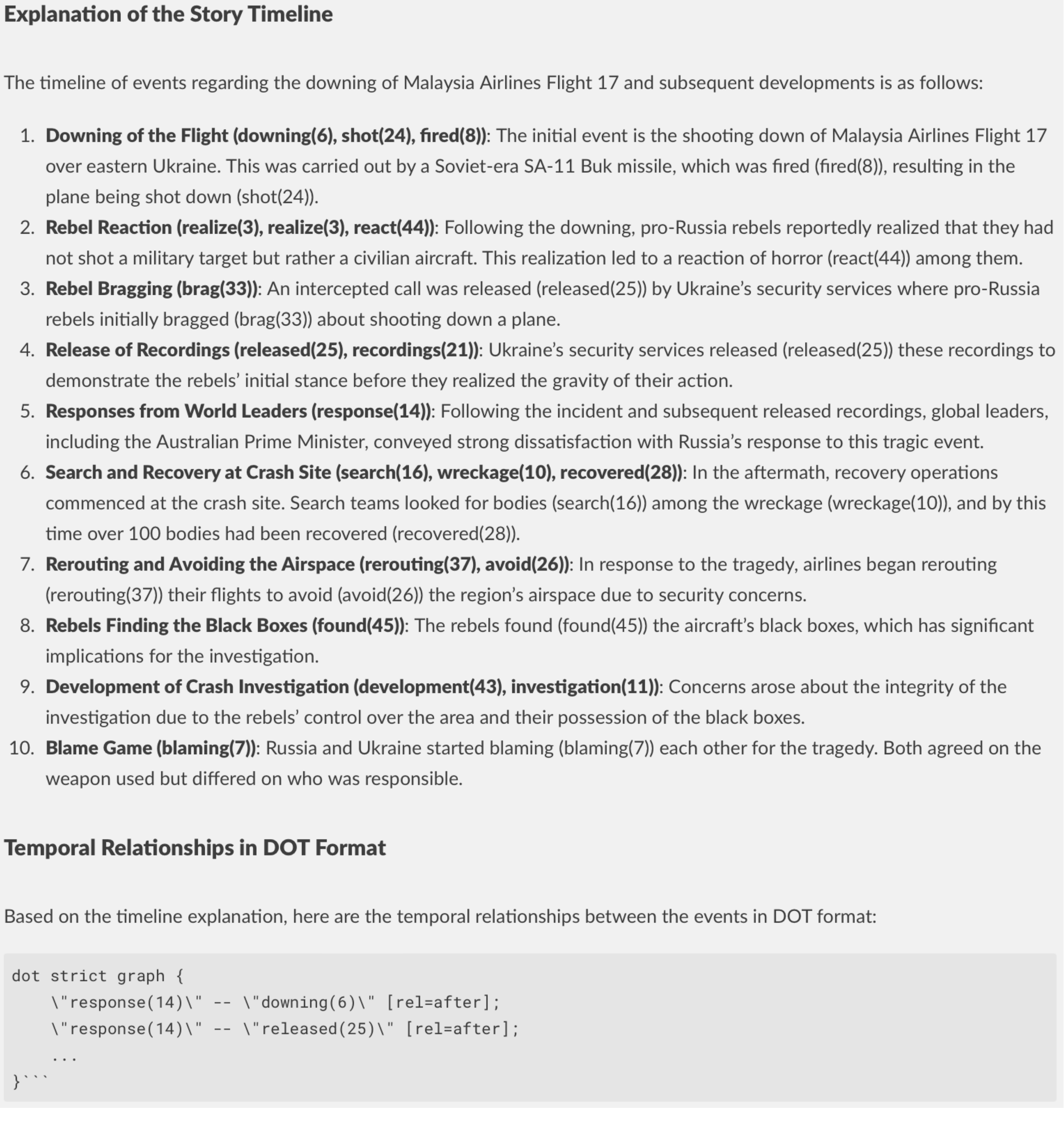
```
strict graph {
"Event1(id)" -- "Event2(id)" [rel=LABEL];
"Event1(id)" -- "Event3(id)" [rel=LABEL];
...
}
```

– Otto Warmbier is dead less than a week after <returning(39)> to the US from North Korea <The rest of the text is omitted to fit in Figure>...
Pairs require classification:
returning(39) -- detention(14)
returning(39) -- lived(3)
<The rest of the relations are omitted to fit in Figure>...

Figure 6: An example of the ZSL-Global prompt.

	MATRES	TB-Dense	TCR	TDD-Manual	NarrativeTime	**OmniTemp**
			Datasets Statistics			
Documents	275	36	25	34	36	30
Events	6,099	1,498	1,134	1,101	1,715	470
before	6,852 (50)	1,361 (21)	1,780 (67)	1,561 (25)	17,011 (22)	1,540 (44)
after	4,752 (35)	1,182 (19)	862 (33)	1,054 (17)	18,366 (23)	1,347 (39)
equal	448 (4)	237 (4)	4 (0)	140 (2)	5,298 (7)	150 (4)
vague	1,525 (11)	2,837 (45)	–	–	25,679 (33)	446 (13)
includes	–	305 (5)	–	2,008 (33)	5,781 (7)	–
is-included	–	383 (6)	–	1,387 (23)	6,639 (8)	–
overlaps	–	–	–	–	227 (0)	–
Total Relations	13,577	6,305	2,646	6,150	79,001	3,483
			Per Document Average Annotation Sparsity			
Events	22.2	41.6	45.4	32.4	47.6	15.6
Actual Relations	49.4	183.7	105.8	180.9	1,110.1	114.9
Expected Relations	234.8	844.5	1,006.1	508.1	1,110.1	114.9
Missing Relations	79%	78.3%	89.5%	64.4%	0%	0%

Table 6: The upper part of the table presents the statistics of notable datasets for the temporal relation extraction task alongside OmniTemp. In parentheses, the values indicate the percentage of each relation type relative to the total relations in the dataset. The bottom part of the table summarizes the average percentage of missing relations per document, calculated as the ratio of actual annotated relations to a complete relation coverage, referred to as *Expected Relations*.



Explanation of the Story Timeline

The timeline of events regarding the downing of Malaysia Airlines Flight 17 and subsequent developments is as follows:

1. **Downing of the Flight (downing(6), shot(24), fired(8))**: The initial event is the shooting down of Malaysia Airlines Flight 17 over eastern Ukraine. This was carried out by a Soviet-era SA-11 Buk missile, which was fired (fired(8)), resulting in the plane being shot down (shot(24)).
2. **Rebel Reaction (realize(3), realize(3), react(44))**: Following the downing, pro-Russia rebels reportedly realized that they had not shot a military target but rather a civilian aircraft. This realization led to a reaction of horror (react(44)) among them.
3. **Rebel Bragging (brag(33))**: An intercepted call was released (released(25)) by Ukraine's security services where pro-Russia rebels initially bragged (brag(33)) about shooting down a plane.
4. **Release of Recordings (released(25), recordings(21))**: Ukraine's security services released (released(25)) these recordings to demonstrate the rebels' initial stance before they realized the gravity of their action.
5. **Responses from World Leaders (response(14))**: Following the incident and subsequent released recordings, global leaders, including the Australian Prime Minister, conveyed strong dissatisfaction with Russia's response to this tragic event.
6. **Search and Recovery at Crash Site (search(16), wreckage(10), recovered(28))**: In the aftermath, recovery operations commenced at the crash site. Search teams looked for bodies (search(16)) among the wreckage (wreckage(10)), and by this time over 100 bodies had been recovered (recovered(28)).
7. **Rerouting and Avoiding the Airspace (rerouting(37), avoid(26))**: In response to the tragedy, airlines began rerouting (rerouting(37)) their flights to avoid (avoid(26)) the region's airspace due to security concerns.
8. **Rebels Finding the Black Boxes (found(45))**: The rebels found (found(45)) the aircraft's black boxes, which has significant implications for the investigation.
9. **Development of Crash Investigation (development(43), investigation(11))**: Concerns arose about the integrity of the investigation due to the rebels' control over the area and their possession of the black boxes.
10. **Blame Game (blaming(7))**: Russia and Ukraine started blaming (blaming(7)) each other for the tragedy. Both agreed on the weapon used but differed on who was responsible.

Temporal Relationships in DOT Format

Based on the timeline explanation, here are the temporal relationships between the events in DOT format:

````
dot strict graph {
    \"response(14)\" -- \"downing(6)\" [rel=after];
    \"response(14)\" -- \"released(25)\" [rel=after];
    ...
}```
````

Figure 7: An example of a generated output when GPT-4o is prompted using the ZSL-Timeline method (with the Markdown format retained from the original output). The full event list is generated; however, it is trimmed (indicated by "...") in this example to ensure the output fits within the figure.

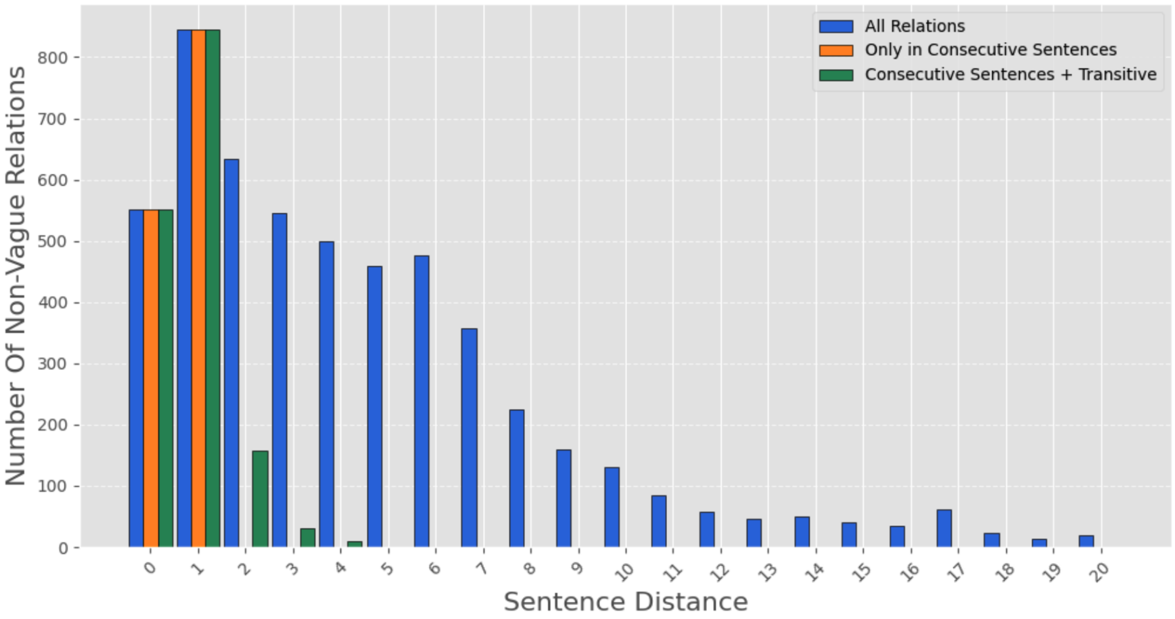


Figure 8: Illustration of the achieved relation distance after applying transitive closure in resources annotated only between consecutive sentences. The blue bars represent the original set of relations in NarrativeTime, which is exhaustively annotated between all events. The orange bars represent the version created by considering only relations between events in consecutive sentences. The green bars represent the set of relations after applying a transitive algorithm to infer additional relations.