More than Classification: A Unified Framework for Event Temporal Relation Extraction

Quzhe Huang[1,2], **Yutong Hu**[1,2], **Shengqi Zhu**[3],
Yansong Feng[1*], **Chang Liu**[1,4], **Dongyan Zhao**[1,5]
[1]Wangxuan Institute of Computer Technology, Peking University, China
[2]School of Intelligence Science and Technology, Peking University
[3]University of Washington
[4]Center for Data Science, Peking University
[5]National Key Laboratory of General Artificial Intelligence
{huangquzhe,huyutong,fengyansong,liuchang97,zhaody}@pku.edu.cn
sqzhu@uw.edu

Abstract

Event temporal relation extraction (ETRE) is usually formulated as a multi-label classification task, where each type of relation is simply treated as a one-hot label. This formulation ignores the meaning of relations and wipes out their intrinsic dependency. After examining the relation definitions in various ETRE tasks, we observe that all relations can be interpreted using the start and end time points of events. For example, relation *Includes* could be interpreted as event 1 starting no later than event 2 and ending no earlier than event 2. In this paper, we propose a unified event temporal relation extraction framework, which transforms temporal relations into logical expressions of time points and completes the ETRE by predicting the relations between certain time point pairs. Experiments on TB-Dense and MATRES show significant improvements over a strong baseline and outperform the state-of-the-art model by 0.3% on both datasets. By representing all relations in a unified framework, we can leverage the relations with sufficient data to assist the learning of other relations, thus achieving stable improvement in low-data scenarios. When the relation definitions are changed, our method can quickly adapt to the new ones by simply modifying the logic expressions that map time points to new event relations. The code is released at https://github.com/AndrewZhe/A-Unified-Framework-for-ETRE.

1 Introduction

In order to fully understand natural language utterances, it is important to understand the temporal information conveyed in the text, especially the relations between the events (Pustejovsky et al., 2003a, 2010). Such temporal relations play an essential role in downstream applications, such as question answering, event timeline generation, and information retrieval (Choubey and Huang, 2017; Han et al., 2019). The Event Temporal Relation Extraction (ETRE) task is proposed to address the extraction of temporal relations between event pairs from text.

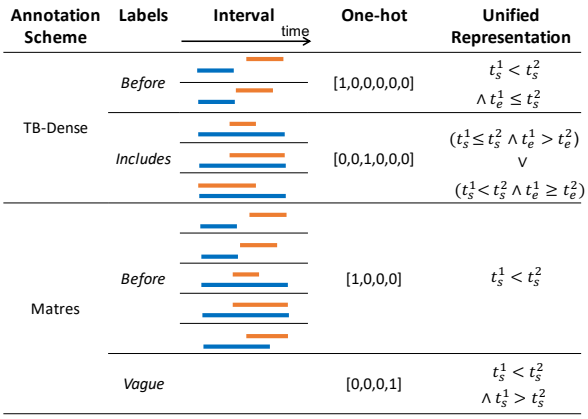


Figure 1: Examples of labels from TB-Dense and MATRES and their Interval, One-hot and Unified representations. — and — represent the intervals of event 1 and event 2 in the timeline. t_s^* and t_e^* represent the start and end time points of an event.

Researchers have different ideas on how to define the temporal relations between two events. Allen (1981) treats an event as an interval in the timeline and uses 13 relations between two intervals to define the temporal relations between events. The 13 relations, together with a special relation *Vague*, are then adopted by TimeBank (Pustejovsky et al., 2003b). However, such a definition is so fine-grained that some relations are very hard to distinguish from each other. Thus following works make a simplification and aggregate some relations (Uz-Zaman et al., 2013; Styler IV et al., 2014). For example, TB-Dense (Cassidy et al., 2014) aggregates *Before* and *Before Immediately* into one coarse re-

* Corresponding author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 9631–9646

lation *Before*. Other studies, like MATRES (Ning et al., 2018), think that identifying the duration of events requires very long contexts and even commonsense, making it exceedingly difficult to determine when an event ends. Therefore, in MATRES, only the start time of events is considered for temporal relations. We show some examples of temporal relations and their interval representations in Figure 1. It can be seen that despite the differences across definitions, each relation reflects certain aspects of the full temporal relationship and has rich meanings behind a single label.

Although the meaning of a relation is important, previous studies did not pay enough attention. They solve ETRE as a simple text classification task, first using an encoder to get the event pair representation and then feeding it into a multi-layer perceptron to get the prediction. All efforts are focused on generating better event pair representations, such as pre-training a task-specific language model (Han et al., 2020) or applying Graph Neural Networks to incorporate syntactic information (Zhang et al., 2022). However, the relations are only used as one-hot labels to provide guidance in cross-entropy loss. Such a classification-based method cannot fully use the meaning of relations and could cause the following problems:

Misunderstanding Relations: Some relations may correspond to complex scenarios, such as *Vague* in MATRES. It describes a contradictory situation where event 1 may occur before event 2 and event 2 may also occur before event 1. Such complex meaning cannot be conveyed by a simple one-hot vector.

Missing the Dependency: The classification-based method treats different relations as orthogonal vectors, however, relations within the same task definition are not independent. For example, both the relations *Includes* and *Before* in TB-Dense imply that event 1 does not start later than event 2.

Lacking Generalization: Since there is no one-to-one mapping between different relation definitions, the classification-based method needs training a unique classifier for every definition. For example, the relation *Includes* in TB-Dense contains three interval relations, and only two of them overlap with relation *Before* in MATRES. Therefore, when a classifier trained on TB-Dense predicts *Includes*, it cannot figure out which relation it should predict under the definition of MATRES.

To address the aforementioned issues, we need a unified framework that can interpret any single relation and connect different ones. We go back to Allen's interval theory, and notice that the relation between intervals is determined by their endpoints, which represent the start and end time points of events. As nearly all definitions of ETRE are based on Allen's interval representation, we find that we can use the relation among the start and end time points of events to represent the relations in any definitions. As illustrated in Figure 1, *Includes* in TB-Dense could be represented as $(t_s^1 \leq t_s^2 \wedge t_e^1 > t_e^2) \vee (t_s^1 < t_s^2 \wedge t_e^1 \geq t_e^2)$.

Inspired by this finding, we design a unified temporal relation extraction framework based on the time points of events. Specifically, based on the relation definitions, we first transform each relation into a logical expression of time point relations, as shown in the last column of Figure 1. Then the task of predicting the temporal relation between events becomes the task of predicting the relation of time points. Following the annotation guidelines by Ning et al. (2018), we infer the relation between two time points t1 and t2 by asking the model two questions: 1) whether t1 could occur earlier than t2 and 2) whether t2 could occur earlier than t1. By answering these questions, we can deepen the association of different time point relations.

Our experiments show that the unified framework can significantly help temporal relation extraction, compared to a strong baseline, and outperforms state-of-the-art (SOTA) model by 0.3% F1 on both TB-Dense and MATRES. By using time points to explicitly interpret the relations, we help the model to better understand ambiguous relations such as *Vague*, and significantly reduce the number of instances misclassified as *Vague*. In addition, since different relations can all be represented as logic expressions of the same time points, we can capture the dependency between different relations. The relations with more training data can be used to assist the learning of relations with fewer data, thus achieving stable improvement in low-data scenarios. When the definitions of temporal relations are changed, we can easily adapt to the new ones by modifying the logic expressions that map time points to new event relations. Further experiments with ChatGPT[1] show that our unified framework can also help Large Language Models(LLMs), outperforming classification-based prompts by 2.3% F1 on TB-Dense.

[1]https://chat.openai.com/

2 Problem Formulation

Given an input sequence $\mathbf{X}$ with two events e_1 and e_2, the task of event temporal relation extraction is to predict a relation from $\mathcal{R} \cup \{Vague\}$ between the event pair (e_1 and e_2), where $\mathcal{R}$ is a pre-defined set of temporal relations of interests. Label *Vague* means the relation between the two events can not be determined by the given context.

3 Enhanced Baseline Model

We first introduce our baseline model for ETRE. It is based on a strong entity relation extraction model (Zhong and Chen, 2021) and we integrate other techniques to make it suitable in ETRE. Our baseline can achieve comparable or even better performance with the previous SOTA in ETRE, providing a powerful encoder for our unified framework.

3.1 Event Encoder

Given two event mentions (e_1, e_2) and a text sequence $\mathbf{X} = [x_1, x_2, ..., x_n]$ of n tokens, the event encoder aims to calculate the representation of the event pair. Considering cross-sentence information has been proven useful in entity relation extraction (Wadden et al., 2019), we believe ETRE will also benefit from it. Thus we extend the input text with 1 more sentence from the left and right context of the sentence containing mentions. To highlight the event mentions, we insert event markers <EVENT_1>, </EVENT_1>, <EVENT_2> and </EVENT_2> into the sequence $\mathbf{X}$ before and after the two events. The new sequence with text markers inserted is then fed into a pre-trained language model, and we use the contextual embeddings of <EVENT_1> and <EVENT_2>, denoted as $\mathbf{h}_{e_1}$ and $\mathbf{h}_{e_2}$ respectively, to calculate the representation of the event pair:

$$\mathbf{ee} = [\mathbf{h}_{e_1} \oplus \mathbf{h}_{e_2}] \quad (1)$$

where $[* \oplus *]$ is the concatenation operator.

3.2 Classifier

Following previous efforts in ETRE (Wen and Ji, 2021; Zhang et al., 2022), our baseline uses a multilayer perceptron (MLP) and a softmax layer to convert the representation of the event pair into a probability distribution:

$$P(\mathbf{R}|e_1, e_2) = \text{softmax}(\mathbf{MLP}(ee)) \quad (2)$$

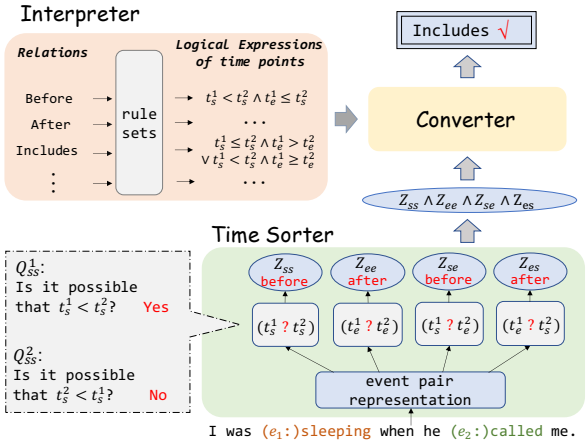


Figure 2: Architecture Overview

where $P(\mathbf{R_i}|e_1, e_2)$ denotes the probability of relation i exiting between e_1 and e_2. We use the cross entropy loss for training.

3.3 Label Symmetry

Inspired by Zhang et al. (2022); Hwang et al. (2022), based on the symmetry property of temporal relations, we expand our training set using rules provided in Appendix B while keeping the validation set and test set unchanged.

4 Our Unified Framework

Generally, going through the classifier, a model can easily output a probability for each category. However, it is difficult for a model to understand what a category represents from the one-hot formed supervision signals, and the model will struggle in summarizing the category's meaning from training data. Meanwhile, since the categories, which actually are relations, are treated as orthogonal signals for supervision, the data of a particular relation cannot help the model understand other relations.

To help the model make better use of the temporal information embedded in relations, we transform the task of predicting the temporal relations into a judgment of the relationship between the start and end time points of two events, which are the basic elements that make up temporal relations in different ETRE definitions.

As shown in Figure 2, our unified framework is composed of three parts: the first is the interpreter, which translates each relation into a logical expression of time points; the second part is the temporal predictor, which predicts the relation between time points based on the representation of

Relation	Unified Rep	$F^{Q\to R}$
Before	$t_e^1 \le t_s^2$	$\neg Q_{es}^2$
After	$t_s^1 \ge t_e^2$	$\neg Q_{se}^1$
Includes	$(t_s^1 \le t_s^2 \wedge t_e^1 > t_e^2)$ $\vee$ $(t_s^1 < t_s^2 \wedge t_e^1 \ge t_e^2)$	$(\neg Q_{ss}^2 \wedge \neg Q_{ee}^1 \wedge Q_{ee}^2)$ $\vee (Q_{ss}^1 \wedge \neg Q_{ss}^2 \wedge \neg Q_{ee}^1)$
Included In	$(t_s^2 \le t_s^1 \wedge t_e^2 > t_e^1)$ $\vee$ $(t_s^2 < t_s^1 \wedge t_e^2 \ge t_e^1)$	$(\neg Q_{ss}^1 \wedge \neg Q_{ee}^2 \wedge Q_{ee}^1)$ $\vee (Q_{ss}^2 \wedge \neg Q_{ss}^1 \wedge \neg Q_{ee}^2)$
Simultaneous	$t_s^1 = t_s^2 \wedge t_e^1 = t_e^2$	$\neg Q_{ss}^1 \wedge \neg Q_{ss}^2 \wedge \neg Q_{ee}^1 \wedge \neg Q_{ee}^2$
Vague	$(t_s^1 < t_s^2 \wedge t_s^1 > t_s^2)$ $\vee (t_e^1 < t_e^2 \wedge t_e^1 > t_e^2)$ $\vee (t_s^1 < t_e^2 \wedge t_s^1 > t_e^2)$ $\vee (t_e^1 < t_s^2 \wedge t_e^1 > t_s^2)$	$(Q_{ss}^1 \wedge Q_{ss}^2)$ $\vee (Q_{ee}^1 \wedge Q_{ee}^2)$ $\vee (Q_{se}^1 \wedge Q_{se}^2)$ $\vee (Q_{es}^1 \wedge Q_{es}^2)$

Figure 3: Relations in TB-Dense, and their unified representations and the logical expressions from Q to R($F^{Q\to R}$)

an event pair; finally, the converter checks which logical expression is satisfied with the assignments from the second stage and thus infer the relations between two events.

4.1 Interpreter

Following Allen's theory, the events e_1 and e_2 could be represented as two intervals $[t_s^1, t_e^1]$ and $[t_s^2, t_e^2]$, where t_s^* and t_t^* are the start and end times of an event. The event temporal relation then could be represented as the relation between intervals, which is determined by the endpoints of intervals, t_s^1, t_e^1, t_s^2 and t_e^2, for example, the interval relation *Before* could be represented as $t_s^1 < t_e^1 < t_s^2 < t_e^2$. Considering the start time of an event should not be later than its end time, to infer the interval's relation, we only need to consider the relations between four pairs of time points, which are $Z_{ss}(t_s^1, t_s^2)$, $Z_{ee}(t_e^1, t_e^2)$, $Z_{se}(t_s^1, t_e^2)$ and $Z_{es}(t_e^1, t_s^2)$. We show all the 13 interval relations and their time point representations in Appendix A.

Current definitions of temporal relations, such as TB-Dense and Matres, are built up by aggregating some interval relations into one to form a coarse-grained relation set. As they are based on Allen's interval theory, we can also use the time points of t_s^1, t_e^1, t_s^2, and t_e^2 to represent their coarse-grained relations. For example, the relation *Includes* in TB-Dense could be interpreted as $(t_s^1 \le t_s^2 \wedge t_e^1 > t_e^2) \vee (t_s^1 < t_s^2 \wedge t_e^1 \ge t_e^2)$.

The interpreter contains a set of rules to transform the definition of relations into the logical expressions of start and end time points of events. Figure 3 shows the logical expressions of every relation in TB-Dense[2]. The logical expressions of different relations are guaranteed to be mutually exclusive, as long as the two relations do not overlap with each other.

[2]We shows that of MATRES in Appendix C

4.2 Time Point Sorter

There are four relations between two time points, *before*, *after*, *equal*, and *vague*. We could treat it as a four-label classification and use a MLP and a softmax layer to complete the prediction. However, such a method also treats each relation as orthogonal labels and cannot interpret the complex relation, *vague*. Inspired by the annotation guidance in MATRES (Ning et al., 2018), we ask the model to answer the following two questions to decide the relation between two time points t^1 and t^2: Q^1: Is it possible that t^1 occur earlier than t^2? and Q^2: Is it possible that t^2 occur earlier than t^1? The model only needs to answer *yes* or *no* to these two questions and the time point relation could be inferred by the rules in Table 1.

Q^1	yes	no	no	yes
Q^2	no	yes	no	yes
Z	*before*	*after*	*equal*	*vague*

Table 1: Mapping from Q to Z

On the one hand, it makes a clear definition of relations like *vague*, which helps the model understand such relations. On the other hand, the dependency between time point relations could be reflected in the same answer to one question, e.g., Q^2 for both relation *before* and *equal* is *no*, which means it is impossible that t^2 is earlier than t^1 in both of these two relations.

To obtain the answers for **Q**'s, we use a two-layer perceptron to simulate the procedure of answering the questions:

$$logit_{tp}^i = FFN_{tp}^2(\sigma(FFN_{tp}^1(\mathbf{ee}))) \quad (3)$$

$$P(Q_{tp}^i) = \text{sigmoid}(\frac{logit_{tp}^i}{\tau}) \quad (4)$$

$$Q_{tp}^i = \mathbb{1}\{P(Q_{tp}^i) > 0.5\} \quad (5)$$

where time point pair $tp \in \{ss, ee, se, es\}$, $i \in \{1, 2\}$, $\mathbb{1}$ denotes the indicator function, $\text{sigmoid}(\frac{*}{\tau})$ is a sigmoid function with temperature τ used to control the smoothing degree of the probability distribution, $P(Q_{tp}^i)$ denotes the probability of answering *yes* for the question i of time point pair tp and Q_{tp}^i is the binary answer, 1 for *yes* and 0 for *no*.

4.3 Converter

After predicting the value of $\mathbf{Z}$, that is, we have obtained the relations between the start and end time points of two events, we need to check which logical expression in the interpreter is True under this set of assignments. As relations are exclusive to each other, we will find only one logical expression with True value and the relation corresponding to this expression will be the temporal relation between the events.

4.4 Inference

As discussed above, the mapping from $\mathbf{Q}$ to $\mathbf{Z}$ and the mapping from $\mathbf{Z}$ to $\mathbf{R}$ could both be represented as logical expressions. Thus, we could also use a logical expression of $\mathbf{Q}$ to directly represent the relations between events, which is denoted as $F^{\mathbf{Q}\to\mathbf{R}}$. Table 3 shows the logical expressions of all relations in TB-Dense[3].

4.5 Training with Soft Logic

So far, we have discussed how to use hard logic to infer the event relation $\mathbf{R}$ from the values of $\mathbf{Q}$. However, in practice, the hard logic reasoning procedure is not differentiable. We thus use soft logic (Bach et al., 2017) to encode the logic expressions from $\mathbf{Q}$ to $\mathbf{R}$. Specifically, soft logic allows continuous truth values from the interval [0, 1] instead of {0,1} (Hu et al., 2016), and the Boolean logic operators are reformulated as:

$$
\begin{aligned}
a \wedge b &= a \cdot b \\
a \vee b &= a + b - a \cdot b \\
\neg a &= 1 - a
\end{aligned}
$$

where $\wedge$ and $\vee$ are approximations to logical conjunction and disjunction.

We substitute the Boolean operators in $F^{\mathbf{Q}\to\mathbf{R}}$ with soft logic operators to get a differentiable mapping from Q to R, $F^{\mathbf{Q}\to\mathbf{R}}_{soft}$, and the probability of $\mathbf{R}$ can be formed as:

$$
P(\mathbf{R}) = F^{\mathbf{Q}\to\mathbf{R}}_{soft}(P(\mathbf{Q})) \tag{6}
$$

$$
P(\mathbf{Q}) = \{P(Q^i_{tp}) | tp \in \{ss, ee, se, es\}, i \in \{1, 2\}\}
$$

where $P(Q^i_{tp})$ is calculated by Equation 4. With the probability of $\mathbf{R}$, we can use the normal cross-entropy loss function to train our model.

[3]The relations of MATRES are shown in Appendix C

5 Experiments

Datasets We conduct experiments over two temporal relation extraction benchmarks, TB-Dense (Cassidy et al., 2014) and MATRES (Ning et al., 2018), both of which could be used for research purposes. TB-Dense includes 6 types of temporal relations: *Before*, *After*, *Includes*, *Is_Included*, *Simultaneous* and *Vague*. Temporal relations in MATRES are annotated only based on start time points, reducing them to 4 types: *Before*, *After*, *Equal* and *Vague*. we use the same train/dev/test splits as previous studies (Han et al., 2019; Wen and Ji, 2021).

Evaluation Metrics For fair comparisons with previous research, we adopt the same evaluation metrics as Zhang et al. (2022). On both TB-Dense and MATRES, we exclude the *Vague* label and compute the micro-F1 score of all the others.

5.1 Main Results

Table 2 reports the performance of our unified framework and baseline methods on TB-Dense and MATRES. Overall, applying our unified framework brings significant improvements compared with Enhanced-Baseline, and outperforms previous SOTA by 0.3% F1 on both datasets, respectively. The only difference between ours and Enhanced-Baseline is that we use the events' start and end points to infer the relation and Enhanced-Baseline directly predicts the relation. The stable improvements on both benchmarks indicate our unified framework could help the model better understand temporal relations.

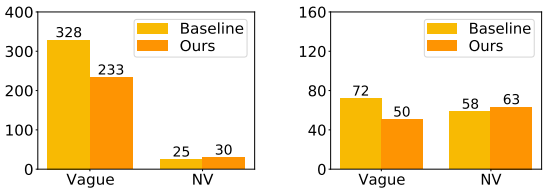
Compared to the improvements on MATRES, which are 0.8% and 0.7% for BERT-Base and RoBERTa-Large respectively, we have a more significant gain on TB-Dense, which is nearly 2% for both base and large models. This is because MATRES only cares about the start time of events and thus cannot benefit from our interpreter module. The improvements on MATRES show that in time point sorter, splitting the decision of time point relations into answering Q_1 and Q_2 is effective. And the greater improvements in TB-Dense further illustrate the usefulness of the interpreter module.

In addition, we notice consistent gains with either BERT-Base or RoBERTa-Large as the backbone. On TB-Dense, our method outperforms Enhanced-Baseline with about 2% F1 for both BERT-Base and RoBERTa-Large, and the gain is

Model	Pretrained Model	TB-Dense	MATRES
LSTM (Cheng and Miyao, 2017)	BERT-Base	62.2	73.4
CogCompTime2.0 (Ning et al., 2019)	BERT-Base	-	71.4
HNP (Han et al., 2019)	BERT-Base	64.5	75.5
Box (Hwang et al., 2022)	RoBERTa-Base	-	77.3
Syntactic (Zhang et al., 2022)*	BERT-Base	66.7	79.3
Enhanced-Baseline	BERT-Base	64.8 ± 0.85	78.5 ± 0.69
Unified-Framework (Ours)	BERT-Base	66.4 ± 0.40	79.3 ± 0.45
PSL (Zhou et al., 2021)	RoBERTa-Large	65.2	-
HMHD (Wang et al., 2021)	RoBERTa-Large	-	78.8
DEER (Han et al., 2020)	RoBERTa-Large	66.8	79.3
Time-Enhanced (Wen and Ji, 2021)	RoBERTa-Large	-	81.7
HGRU (Tan et al., 2021)	RoBERTa-Large	-	80.5
Syntactic (Zhang et al., 2022)*	BERT-Large	67.1	80.3
SCS-EERE (Man et al., 2022)	RoBERTa-Large	-	81.6
TIMERS (Mathur et al., 2021a)	BERT-Large	67.8	82.3
Enhanced-Baseline	RoBERTa-Large	66.2 ± 2.08	81.9 ± 0.35
Unified-Framework (Ours)	RoBERTa-Large	68.1 ± 1.35	82.6 ± 1.05

Table 2: F1 score on TB-Dense and MATRES. Models marked * use additional training resources. SCS-EERE only reports the maximum score among multi-experiments. We re-run the code provided in the original work and report the average F1 among 3 experiments here.

about 1% for both these two backbones on MATRES. The consistent improvement implies that the efficacy of our unified framework is orthogonal to the encoders' capability. We evaluate our methods with a very strong encoder, whose baseline version is comparable with the SOTA in MATRES, to show the benefits of using a unified framework will not disappear with the development of strong encoders. We believe, in the future, with better event pair representations, e.g., incorporating syntactic information like Zhang et al. (2022), our framework would remain effective.



(a) TB-Dense (b) MATRES

Figure 4: Incorrect cases number of baseline (Enhanced-Baseline) and our model on both TB-Dense and MATRES.

6 Analysis

We further explore how our module makes better use of label information in ETRE tasks. We show the 3 problems of classification-based methods, mentioned in Introduction, could be alleviated by our unified framework.

6.1 Better Comprehension of Relations

For classification-based methods, every label is treated as a one-hot vector, and models have to guess these vectors' meanings through training data. In contrast, we interpret every temporal relation into a logical expression of start and end time points, which clearly states the meaning of the relation. Among all the temporal relations between two events, *Vague* is the most ambiguous one, because it does not describe a specific situation and it could correspond to various possibilities. Therefore, it is very hard for a model to summarize this relation's meaning from training data. We will show using logical expressions to make clear definitions, could benefit ambiguous relations, like *Vague*.

We focus on the positive instances whose gold label is not *Vague*, and Figure 4 shows the number of instances misclassified as relation *Vague*, and the number of instances misclassified as others, which is denoted as NV. We can see that *Vague*-related errors make up the majority, which reflects the challenge posed by the ambiguity of *Vague*. Comparing the performance of the baseline and ours, we see that the number of errors associated with *Vague* decreases by 95 and 22 in TB-Dense and MATRES, respectively. This significant decrease indicates that by explicitly interpreting the meaning of *Vague* using a logical expression, our approach

		1%	5%	10%	20%	30%	Avg
Mi-F1	Base	28.8	47.1	51.4	57.9	60.3	
	Ours	29.2	50.8	56.5	60.6	62.8	
	Δ	+0.4	+3.7	+5.1	+2.7	+2.5	+2.9
Ma-F1	Base	13.8	25.7	32.2	37.0	39.5	
	Ours	16.6	27.7	33.0	38.6	41.1	
	Δ	+2.8	+2.0	+0.8	+1.6	+1.6	+1.8

Table 3: Performance of baseline (Enhanced-Baseline) and our model in low-data scenarios. We use 1%, 5%, 10%, 20% and 30% data in TB-Dense to train a BERT-Base based model. Mi-F1 and Ma-F1 represent Micro-F1 and Macro-F1, respectively.

Model	Normal	Transfer
Baseline(Mapping1)	81.9 ± 0.35	63.1 ± 1.1
Baseline(Mapping2)	81.9 ± 0.35	64.3 ± 0.7
Ours	82.6 ± 1.05	70.4 ± 0.8

Table 4: Results of the transfer learning experiment. Normal indicates models are trained on MATRES and tested on MATRES. Transfer indicates models are trained on TB-Dense and tested on MATRES.

can help the model better understand this relation and alleviate the confusion between this relation and others. There is a slight increase of errors not related to *Vague* on both datasets. These errors are mainly related to *Before* and *After*, whose meaning is not so ambiguous and thus may not benefit from our approach.

6.2 Capability of Capturing Dependency

Classification-based methods treat relations as independent labels and thus the instance of one relation could not help the model to understand other relations. Different from such methods, we represent all temporal relations in a unified framework and different relations could be connected via specific time point pairs. For example, *Before* and *Includes* in TB-Dense share similar relations between the start points of two events, which is $t_s^1 \leq t_s^2$. Thanks to such connections, when a model meets an instance whose relation is *Before*, the model could also learn something about *Includes*. This enables the model to leverage relations with sufficient training data to aid in the understanding of relations whose training data is limited.

We show our method could improve the efficiency of data utilization by analyzing the performance in low-data scenarios. Due to the unbalanced label distribution, relations like *Includes* have very few training samples in low-data scenarios and thus it would be hard to learn by itself. We randomly sample 1%, 5%, 10%, 20%, and 30% cases from TB-Dense, and Table 3 shows the performance of the baseline and our method.

Overall, our method achieves a stable improvement compared to the baseline in all settings. On average, our method outperforms the baseline by 2.9% and 1.8% for micro-F1 and macro-F1, respectively. This shows that our method is capable of using data more effectively. As shown in Table 2, our method improves 1.9% micro-F1 compared to the baseline when trained with the whole TB-Dense, which is lower than the average improvement under low resources, indicating that our method has more potential in low resource scenarios. We note that in the scenario with the smallest amount of training data, i.e., setting 1%, the difference of micro-F1 between ours and the baseline is relatively small. This is because, in this scenario, there are few instances corresponding to the relations *Includes*, *Is_Included* and *Equal*, and the baseline model directly degenerates into a 3-way classifier, only predicting *Before*, *After* and *Vague*. As *Before* and *After* also account for most of the test sets, the baseline achieves a good micro-F1. Our method, on the other hand, is capable of learning relations like *Includes*, which has limited training samples, through relations with sufficient data, like *Before*. The good comprehension of relations with limited data is demonstrated by the significant improvement on macro-F1, where our method outperforms the baseline by 2.8%.

6.3 Adaptation to Different Definitions

One advantage of modeling the relation between time points instead of directly predicting the relation between events is that our method can be adapted to different task definitions. The relations in different task definitions, though, have different meanings, all of them could be interpreted using the relation between time points. For example, TB-Dense and MATRES have different relation definitions, but we could learn how to determine the relation between t_s^1 and t_s^2 from TB-Dense and then use this kind of time point relation to directly infer the temporal relations in MATRES. In other words, we only need to modify the logic expressions that map the time point relations to the event relations, when the task definitions are changed, and do not have to train a new model from scratch.

The situation is quite different for methods that directly predict the relationships between events. This is because there is not a strict one-to-one mapping between different task definitions. One typical example is the relation *Vague* in TB-Dense. It might indicate the relation between the start time of the two events is uncertain or the two events start in a determined order but it is hard to determine which event ends first. In this case, the *Vague* in TB-Dense may correspond to all four relations in MATRES. Another example is that *Includes* in TB-Dense indicates that the start time of event 1 is no later than event 2, which could be either the *Before*, *Equal*, or *Vague* relation in MATRES.

We evaluate models' ability to adapt to different task definitions by training on TB-Dense and testing on MATRES. For our approach, the time point sorter of t_s^1 and t_s^2 trained on TB-Dense can be directly used to infer the relations in MATRES. And for the baseline model, it is necessary to map the relations in TB-Dense to the relations in MATRES. Firstly, *Before*, *After*, *Simultaneous* and *Vague* in TB-Dense are mapped to *Before*, *After*, *Equal* and *Vague* in MATRES, respectively. Then for the remaining two relations, *Includes* and *Is_Included in*, we apply two different mappings, one is to map them both to *Vague* in MATRES because we could not determine the specific start time relation, which is denoted as Mapping1. The other is to map *Includes* to *Before* and *Is_Included* to *After*, considering the probability of two events starting simultaneously is small, and we denote this as Mapping2.

Table 4 shows the average micro-F1 score and standard deviation of two models using RoBERTa-Large. We can see that, our model outperforms the baseline by 0.7% F1 when trained and tested both on MATRES. As the training set changed from TB-Dense to MATRES, there is a significant increase in the gap between our model and baseline with both two mapping methods. We outperform Mapping1 by 7.3% and outperform Mapping2 by 6.1%, which shows our advantage in transfer learning. By representing all relations in a unified framework, our model bridges the gap between MATRES and TB-Dense, which demonstrates the strong generalization capability of our method.

7 Event RE in LLMs

Large Language Models (LLMs), such as ChatGPT[4], have shown impressive performance in various tasks. In this section, we investigate the performance of LLMs in event temporal relation extraction and assess the value of our proposed unified framework in the era of LLMs.

classification	Given relation candidates: [Before, After, Include, Is included, Simultaneous], which is the temporal relation of e1 with respect to e2? Answer "uncertain" if unsure.
unified framework	1. Which starts first? e1 or e2? 2. Which starts first? e2 or e1? 3. Which ends first? e1 or e2? 4. Which ends first? e2 or e1?

Figure 5: An illustration of the two kinds of prompts we used. The complete prompts can be found in Appendix E. The unified framework asks the model to answer the four questions and deduce the final temporal relation based on the answers.

We conduct experiments using gpt-3.5-turbo-0301[5], and Figure 5 shows the prompts we used. The classification-based prompt lists all the candidate relations and requires the model to select one from them. If the list of relation candidates changes, the prompt should be modified accordingly and the old results will be useless. Unlike the classification-based method, our unified framework is irrelevant to the candidate list. We ask the model to answer four questions and deduce the temporal relation based on the answers, just following what we do on Bert-based models. We explore variations of prompts and the detailed experimental settings can be found in Appendix E. Table 5 shows the results on TBDense, and below are our main findings:

The order of candidates matters in the classification-based prompt. The distribution of temporal relations is highly imbalanced in existing datasets, and we find that putting the majority relation, which is *Before*, at the beginning of the candidate list will significantly affect the performance. For the vanilla classification-based prompt, we randomly sample an order of candidates for different cases. In contrast, *+Before First Order* and *+Before Last Order* use a fixed order, which put *Before* at the beginning or end of the candidate list, respectively.[6] As shown in Tabel 5, compared with the other two orders, putting *Before* at the beginning causes at least 2.7% decline in F1. Further analysis shows that in this scenario, the model is more likely to predict *Before*, making up to 55% of all the predictions.

Chain of thought (CoT) can improve accu-

[4]https://chat.openai.com/

[5]https://platform.openai.com/docs/models/gpt-3-5

[6]Please refer to Figure 8 for details.

	P	R	F1
Classification-Based	28.7	48.9	36.1
+ Before Fist Order	26.7	44.7	33.4
+ Before Last Order	29.9	51.7	37.9
- Relation Direction	29.8	36.4	32.8
+ CoT	31.5	43.9	36.6
+ CoT + Self-Consistency	33.8	45.5	38.7
Unified Framework(Ours)	42.1	39.9	41.0

Table 5: Performance of ChatGPT on TBDense

racy. Instead of directly generating the temporal relation between two events, generate the reasoning procedure first and then deduce the answer could provide 0.5% improvement in F1.

Single word answer might not determine the direction of relations. When the model is expected to return a single word, like *Before* to represent the temporal relation between two events, it might mean e_1 *before* e_2, but it could also mean e_2 *before* e_1. This is a common phenomenon when the prompt does not explicitly mention the direction of relation, e.g., "What is the temporal relation between e_1 and e_2". This would lead to inferior performance, that the vanilla classification-based prompt outperforms the *-Relation Direction Prompt* by 3.3% in F1.

A unified framework may help Large Language Models (LLMs). As shown in Table 5, using our unified framework prompt could achieve 41.0% F1, surpassing all classification-based variants, including the variant that incorporates self-consistency trick (Wang et al., 2022b).

8 Related Work

Earlier studies have proposed various definitions of the relations between two events, and all of them adopt Allen's interval representation. The 13 interval relations together with 1 *Vague* relation form the basis elements for other relation definitions. TimeBank (Pustejovsky et al., 2003b) and TempEval-3 (UzZaman et al., 2013) directly use all of the 14 relations and then the researchers find that some relations are too fine-grained for both humans and models. Thus they simplify the ETRE task by aggregating some relations into a coarse one, e.g., Verhagen et al. (2007) merged all the overlap relations into a single relation, overlap. ISO-TimeML (Pustejovsky et al., 2010) pays attention to a special relation, *contain*, which is composed of three interval relations where one interval is within the other. The focus on relation *contain* influences many followers, like THYME (Styler IV et al., 2014), Richer (O'Gorman et al., 2016), TB-Dense (Cassidy et al., 2014) and MAVEN (Wang et al., 2022a). All these definitions, though differ in various aspects, can convert to intervals relations and thus could be interpreted using the endpoints of intervals. In other words, the different relations under these definitions could be represented in our unified framework. We just use two most widely used definitions, TB-Dense and MATRES, to evaluate our framework, and our framework can be applied to other definitions.

To solve the ETRE, previous efforts often regarded this task as a classification problem, and focus on learning better representations of event pairs, e.g, incorporating syntactic information (Meng et al., 2017; Choubey and Huang, 2017; Zhang et al., 2022) or discourse information (Mathur et al., 2021b) into the encoder. Some studies also try to design auxiliary tasks, like event extraction (Zhang et al., 2022) or relative event time prediction (Wen and Ji, 2021) to further enhance the encoder. Different from their work, we focus on helping the model understand temporal relations better after obtaining the event pair representation from the encoder, and thus our work is orthogonal to them. Recently, Hwang et al. (2022) uses a box embedding to handle the asymmetric relationship between event pairs. However, the box embedding can only handle four types of relations, i.e., *Before*, *After*, *Equal* and *Vague*, and it cannot generalize to more complex relations, like *Includes* in TB-Dense. To solve another task, Cheng and Miyao (2020) also consider start and end times separately. However, Cheng and Miyao (2020) directly uses a classifier to determine the relation between time points and cannot understand the relation *Vague* well.

9 Conclusion

In this paper, we interpret temporal relations as a combination of start and end time points of two events. Using this interpretation, we could not only explicitly convey temporal information to the model, but also represent relations of different task definitions in a unified framework. Our experimental results in TB-Dense and MATRES demonstrate the effectiveness of our proposed method, significantly outperforming previous state-of-the-art models in full data setting and providing large improvements on both few-shot and transfer-learning settings. In the future, we will investigate the potential of our approach in cross-document scenarios.

Acknowledgements

This work is supported in part by National Key R&D Program of China (No. 2020AAA0106600) and NSFC (62161160339). We would like to thank the anonymous reviewers for their helpful comments and suggestions; thank Weiye Chen for providing valuable comments. For any correspondence, please contact Yansong Feng.

Limitations

Due to the limitation of dataset resources, we evaluate our unified model only with TB-Dense and MATRES. Although the experiment results show that our approach can significantly outperform state-of-the-art methods, we still need to experiment on more datasets with various kinds of temporal relations to further prove the generalization capability and robustness of our framework.

References

James F Allen. 1981. An interval-based representation of temporal knowledge. In *Proceedings of the 7th international joint conference on Artificial intelligence-Volume 1*, pages 221–226.

Stephen H Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2017. Hinge-loss markov random fields and probabilistic soft logic. *The Journal of Machine Learning Research*, 18(1):3846–3912.

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 501–506.

Fei Cheng and Yusuke Miyao. 2017. Classifying temporal relations by bidirectional LSTM over dependency paths. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–6, Vancouver, Canada. Association for Computational Linguistics.

Fei Cheng and Yusuke Miyao. 2020. Predicting event time by classifying sub-level temporal relations induced from a unified representation of time anchors. *arXiv preprint arXiv:2008.06452*.

Prafulla Kumar Choubey and Ruihong Huang. 2017. A sequential model for classifying temporal relations between intra-sentence events. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1796–1802, Copenhagen, Denmark. Association for Computational Linguistics.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019. Joint event and temporal relation extraction with shared representations and structured prediction. *arXiv preprint arXiv:1909.05360*.

Rujun Han, Xiang Ren, and Nanyun Peng. 2020. Econet: Effective continual pretraining of language models for event temporal reasoning. *arXiv preprint arXiv:2012.15283*.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420.

EunJeong Hwang, Jay-Yoon Lee, Tianyi Yang, Dhruvesh Patel, Dongxu Zhang, and Andrew McCallum. 2022. Event-event relation extraction using probabilistic box embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 235–244, Dublin, Ireland. Association for Computational Linguistics.

Hieu Man, Nghia Trung Ngo, Linh Ngo Van, and Thien Huu Nguyen. 2022. Selecting optimal context sentences for event-event relation extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11058–11066.

Puneet Mathur, Rajiv Jain, Franck Dernoncourt, Vlad Morariu, Quan Hung Tran, and Dinesh Manocha. 2021a. 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.

Puneet Mathur, Rajiv Jain, Franck Dernoncourt, Vlad Morariu, Quan Hung Tran, and Dinesh Manocha. 2021b. 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.

Yuanliang Meng, Anna Rumshisky, and Alexey Romanov. 2017. Temporal information extraction for question answering using syntactic dependencies in an LSTM-based architecture. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 887–896, Copenhagen, Denmark. 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. 2018. 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.

Tim O'Gorman, Kristin Wright-Bettner, and Martha Palmer. 2016. Richer event description: Integrating event coreference with temporal, causal and bridging annotation. In *Proceedings of the 2nd Workshop on Computing News Storylines (CNS 2016)*, pages 47–56.

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

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003b. The timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK.

James Pustejovsky, Kiyong Lee, Harry Bunt, and Laurent Romary. 2010. Iso-timeml: An international standard for semantic annotation. In *LREC*, volume 10, pages 394–397.

William F. Styler IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C de Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, and James Pustejovsky. 2014. Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143–154.

Xingwei Tan, Gabriele Pergola, and Yulan He. 2021. Extracting event temporal relations via hyperbolic geometry. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8065–8077.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*, pages 75–80.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 5783–5788. Association for Computational Linguistics.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2021. Joint constrained learning for event-event relation extraction.

Xiaozhi Wang, Yulin Chen, Ning Ding, Hao Peng, Zimu Wang, Yankai Lin, Xu Han, Lei Hou, Juanzi Li, Zhiyuan Liu, et al. 2022a. Maven-ere: A unified large-scale dataset for event coreference, temporal, causal, and subevent relation extraction. *arXiv preprint arXiv:2211.07342*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

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.

Shuaicheng Zhang, Qiang Ning, and Lifu Huang. 2022. Extracting temporal event relation with syntactic-guided temporal graph transformer. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2022,*.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61.

Yichao Zhou, Yu Yan, Rujun Han, John 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 *AAAI*.

A Allen's Interval Relations

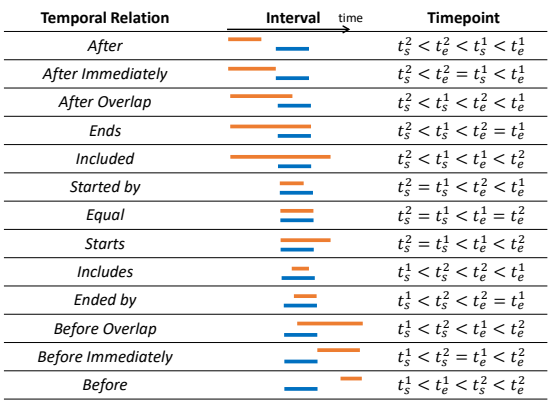


Figure 6: All 13 interval relations defined in Allen (1981). — and — represent the intervals of event 1 and event 2 in the timeline. t_s^* and t_e^* represent the start and end time points of an event.

B Rules of Symmetry

Original Relation	Symmetry Relation
A *Before* B	B *After* A
A *After* B	B *Before* A
A *Include* B	B *Is_included* A
A *Is_included* B	B *Include* A
A *Equal(Simultaneous)* B	B *Equal(Simultaneous)* A
A *Vague* B	B *Vague* A

Table 6: Symmetry rules between temporal relations

C MATRES Relations

Relation	Unified Rep	$F^{Q \to R}$
Before	$t_s^1 < t_s^2$	$Q_{ss}^1 \wedge \neg Q_{ss}^2$
After	$t_s^1 > t_s^2$	$\neg Q_{ss}^1 \wedge Q_{ss}^2$
Equal	$t_s^1 = t_s^2$	$\neg Q_{ss}^1 \wedge \neg Q_{ss}^2$
Vague	$t_s^1 < t_s^2 \wedge t_s^1 > t_s^2$	$Q_{ss}^1 \wedge Q_{ss}^2$

Figure 7: Relations in MATRES, and their unified representations and the logical expressions from Q to R($F^{Q\to}$)

D Implementation Details

For fair comparisons with previous baseline methods, we use the pre-trained BERT-Base and RoBERTa-Large models for fine-tuning and optimize our model with AdamW. We optimize the parameters with grid search: training epoch ∈ 1, 3, 5, 10, learning rate ∈ 2e-5, 1e-5, training batch size 16, temperature in time point sorter ∈ 1, 10. The best hyperparameters for BERT-Base are (1, 2e-5, 10) and the best hyperparameters for RoBERTa-Large are (3, 1e-5, 10). We using one A40 GPU for training.

Pi	Pj	Time Point Relation
e_1	e_2	before
e_1	ϕ	before
ϕ	e_2	before
e_2	e_1	after
ϕ	e_1	after
e_2	ϕ	after
otherwise		vague

Table 7: Mapping of the answers of prompts to the relation of time points. When Pi and Pj refer to Prompt1 and Prompt2 in Figure 9, we can deduce the relation of the start time point. And when Pi and Pj refer to Prompt3 and Prompt4 in Figure 9, we can deduce the relation of the end time point. e_1 and e_2 indicate the possible answer of LLMs: event_1 and event_2. ϕ means the output of LLMs is not in the label set {event_1, event_2}

Start Time	End Time	Temporal Relation
before	before	Before
after	after	After
before	after	Includes
after	before	Included In
otherwise		Vague

Table 8: Mapping from the relation of the start time point and the end time point to the final temporal relation between two events for ChatGPT.

E Experiment Details for LLMs

Figure 9 and 8 shows all the prompts we used in Section 7. For the variants of classification-based prompts, we ask the model to directly output the temporal relation between two events. In Unified Framework, we design four prompts to first determine the relationship between start and end time points and then deduce the final temporal relation. Specifically, we ask LLMs which event starts first with *Prompt1* and *Prompt2* in Figure 9. If the results of the two prompts keep consistent, which means the answers are (event_1, event_2) or (event_2, event_1) for the two prompts respectively, we can determine the temporal relation of

Classification	***Prompt:*** text: *TEXT* event_1: *EVENT_1*, indicated by *** event_2: *EVENT_2*, indicated by ### Give a list of five temporal relationships: **[include, before, is included, after, simultaneous]**. Based on the given text, what is the temporal relation of event_1 with respect to event_2? Answer "uncertain" if unsure. Output the answer with JSON format: {"answer": "certain type of temporal relation from the list, or uncertain"}.
Classification + Before First Order	***Prompt:*** text: *TEXT* event_1: *EVENT_1*, indicated by *** event_2: EVENT_2, indicated by ### Give a list of five temporal relationships: **[before, after, include, is included, simultaneous]**. Based on the given text, what is the temporal relation between event_1 and event_2? Answer "uncertain" if unsure. Output the answer with JSON format: {"answer": "certain type of temporal relation from the list, or uncertain"}.
Classification + Before Last Order	***Prompt:*** text: *TEXT* event_1: *EVENT_1*, indicated by *** event_2: *EVENT_2*, indicated by ### Give a list of five temporal relationships: **[simultaneous, is included, include, after, before]**. Based on the given text, what is the temporal relation of event_1 with respect to event_2? Answer "uncertain" if unsure. Please first describe the reasoning procedure and then output the answer with JSON format: {"answer": "certain type of temporal relation from the list, or uncertain"}.

Figure 8: The details of the designed prompt. In Classification, the order of the temporal relationships is generated randomly. In Classification + Before First Order and Classification + Before Last Order, the order of the temporal relationships is fixed, based on the frequency of relationships in the test dataset. Classification + Before First Order put the most common relation *Before* at the beginning of the list, while Classification + Before Last Order put *Before* in the end.

the start points between the two events. Otherwise, the temporal relation of the start points is set to *Vague*. Sometimes, LLMs may generate answers not in the label set {event_1, event_2}. If both the answers for *Prompt1* and *Prompt2* are not in the label set, we regard the relation as *Vague*. If only one answer for the two prompts is not in the label set, we determine the relation of start time points solely based on the other. We use the same rules to obtain the relation of the end time points based on the answers of *Prompt3* and *Prompt4*. Table 7 shows the mapping of the answers of prompts to the relation of start and end time points, and Table 8 shows how to get the final temporal relation between the two events.

Classification	***Prompt:*** text: *TEXT* event_1: *EVENT_1*, indicated by *** event_2: *EVENT_2*, indicated by ### Give a list of five temporal relationships: [include, before, is included, after, simultaneous]. Based on the given text, what is the temporal relation of event_1 with respect to event_2? Answer "uncertain" if unsure. Output the answer with JSON format: {"answer": "certain type of temporal relation from the list, or uncertain"}.
Classification - Relation Direction	***Prompt:*** text: *TEXT* event_1: *EVENT_1*, indicated by *** event_2: EVENT_2, indicated by ### Give a list of five temporal relationships: [include, before, is included, after, simultaneous]. Based on the given text, **what is the temporal relation between event_1 and event_2?** Answer "uncertain" if unsure. Output the answer with JSON format: {"answer": "certain type of temporal relation from the list, or uncertain"}.
Classification + Chain of Thought	***Prompt:*** text: *TEXT* event_1: *EVENT_1*, indicated by *** event_2: *EVENT_2*, indicated by ### Give a list of five temporal relationships: [include, before, is included, after, simultaneous]. Based on the given text, what is the temporal relation of event_1 with respect to event_2? Answer "uncertain" if unsure. **Please first describe the reasoning procedure** and then output the answer with JSON format: {"answer": "certain type of temporal relation from the list, or uncertain"}.
Unified Framework	***Prompt1:*** text: *TEXT* event_1: *EVENT_1*, indicated by ### event_2: *EVENT_2*, indicated by *** Based on the given text, which event starts first? Please first describe the reasoning procedure and then output the answer with JSON format: {"answer": "event id which starts first"} ***Prompt2:*** text: *TEXT* event_1: *EVENT_2*, indicated by *** event_2: *EVENT_1*, indicated by ### Based on the given text, which event starts first? Please first describe the reasoning procedure and then output the answer with JSON format: {"answer": "event id which starts first"} ***Prompt3:*** text: *TEXT* event_1: *EVENT_1*, indicated by ### event_2: *EVENT_2*, indicated by *** Based on the given text, which event ends first? Please first describe the reasoning procedure and then output the answer with JSON format: {"answer": "event id which ends first"} ***Prompt4:*** text: *TEXT* event_1: *EVENT_2*, indicated by *** event_2: *EVENT_1*, indicated by ### Based on the given text, which event ends first? Please first describe the reasoning procedure and then output the answer with JSON format: {"answer": "event id which ends first"}

Figure 9: The details of the designed prompt. *TEXT* represents the context containing event_1 trigger *EVENT_1* and event_2 trigger *EVENT_2*. The location of *EVENT_1* and *EVENT_2* in the *TEXT* are emphasized by adding makers ### and *** in front of them respectively.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
9

☑ A2. Did you discuss any potential risks of your work?
9

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

5 6

☑ B1. Did you cite the creators of artifacts you used?
5

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
5

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
5

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
5

C ☑ Did you run computational experiments?

5 6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix D

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix D

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
5 6

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.