ConTempo: A Unified Temporally Contrastive Framework for Temporal Relation Extraction

Jingcheng Niu,[1,2,3] **Saifei Liao,**[2,3] **Victoria Ng,**[1] **Simon De Montigny,**[1] **Gerald Penn**[2,3]
[1]Public Health Agency of Canada, [2]University of Toronto, [3]Vector Institute
{niu,liaosaif,gpenn}@cs.toronto.edu
{victoria.ng,simon.demontigny}@phac-aspc.gc.ca

Abstract

The task of temporal relation extraction (TRE) involves identifying and extracting temporal relations between events from narratives. We identify two primary issues with TRE systems. First, by formulating TRE as a simple text classification task where every temporal relation is independent, it is hard to enhance the TRE model's representation of meaning of temporal relations, and its facility with the underlying temporal calculus. We solve the issue by proposing a novel Temporally Contrastive learning model (ConTempo) that increase the model's awareness of the meaning of temporal relations by leveraging their symmetric or anti-symmetric properties. Second, the reusability of innovations has been limited due to incompatibilities in model architectures. Therefore, we propose a unified framework and show that ConTempo is compatible with all three main branches of TRE research. Our results demonstrate that the performance gains of ConTempo are more pronounced, with the total combination achieving state-of-the-art performance on the widely used MATRES and TBD corpora. We furthermore identified and corrected a large number of annotation errors present in the test set of MATRES, after which the performance increase brought by ConTempo becomes more apparent.

1 Introduction

Temporal relation extraction (TRE) is the task of classifying the temporal relations between pairs of events conveyed in narratives (Pustejovsky et al., 2006, 2010). This task is important to the natural language processing (NLP) community because the conception of time is a key component of text comprehension and reasoning. Additionally, it carries practical importance, as TRE is a crucial component of various downstream applications in question answering, information retrieval, and information extraction.

TRE is difficult. Despite its importance and popularity, TRE systems still have relatively poor performance compared to other natural language understanding tasks. TRE is also salient because recent large language models (LLMs) and chat systems have substantially worse performance than smaller, fine-tuned TRE models (Yuan et al., 2023; Huang et al., 2023).

In this paper, we enhance current TRE systems by identifying and addressing two major issue that hinder their performance:

1. It is difficult to understand relation labels. TRE is typically formulated as a multi-label text classification task. The representation of the relation is determined by a neural classifier, which outputs logits for different label classes. These logits are then converted into label probabilities using a softmax layer. As a result, in a TRE model, all relation types are considered independent. However, temporal relations are inherently structured, governed by an underlying temporal calculus (Allen, 1981, 1983). For instance, if event A occurred before event B, it can be inferred that B happened after A. Integrating the meanings of each temporal relation and the principles of the underlying temporal calculus into the neural classifier presents a complex challenge, as there is no straightforward method to do so.

2. The reusability of innovations has been limited. Many of the recent innovations in TRE have not built upon one another, primarily due to the lack of a unified framework (beyond annotation standards) to accommodate them. For example, despite being widely cited and used as a baseline by subsequent systems, the relative time prediction technique (Wen and Ji, 2021) has rarely been applied on top of newer models, due to incompatibilities in model architecture. The evaluation methods employed by earlier systems vary significantly, moreover, which complicates direct comparisons

Findings of the Association for Computational Linguistics: ACL 2024, pages 1521–1533

between models and further hinders the development of future models.

We propose a novel **Tempo**rally **Con**trastive learning method (ConTempo)[1] to enhance the model's representation of the meaning of temporal relations, and its facility with the underlying temporal calculus. The intuition behind ConTempo is very simple. Contrastive learning aims to pull similar (positive) samples closer and push dissimilar (negative) samples apart. We leverage the symmetric and antisymmetric properties of temporal relations to augment a base graph of positive samples for contrastive learning. For antisymmetric relations (A before B), we want to push the relation's representation away from its inverse (B after A). For a symmetric relation (A equals B), we likewise want to pull its representation closer to its inverse (B equals A). Our experimental results indicate that ConTempo is effective at creating distinctive representations for different types of temporal relation pairs and yields a substantial improvement in performance.

Then, building upon ConTempo, we create a unified framework that is compatible with various previous modes of improvement in TRE systems. In particular, we have identified two major threads of innovation in the previous literature on this topic: (1) GNN encoding of event and context information, (2) incorporation of common-sense information, and (3) the soft enforcement of logical constraints. Our experiment shows that ConTempo is compatible with techniques from all three major threads of research and can achieve state-of-the-art performance on major evaluation corpora, including MATRES (Ning et al., 2018b) and TimeBank-Dense (TBD; Cassidy et al., 2014). We nevertheless perceive the need to introduce CleanMATRES, a refined version of MATRES that we created by fixing certain annotation errors.

Among all of these innovations, an ablation study nevertheless shows that ConTempo brings the most significant performance increase. This finding underscores our initial hypothesis: the primary challenge for these models has been their difficulty with making sense of relation labels.

During our development of the unified ComTempo framework, we discovered a large number of annotation errors within MATRES and TBD, two commonly utilized corpora. Because of TBD's dense annotation scheme, the appropriate level of underspecificity cannot be resolved without further investigation. But we are able to provide a thorough re-examination of the MATRES corpus. After correcting these errors, performance improves yet again, and the performance increase brought by ConTempo becomes more apparent.

[1]ConTempo and CleanMATRES are publicly available online: https://github.com/frankniujc/contempo.

Corpus	Relations
MATRES	BEFORE, AFTER, EQUAL, VAGUE
TBD	BEFORE, AFTER, INCLUDES, IS_INCLUDED, EQUAL, VAGUE

Table 1: Temporal relations used by MATRES and TBD.

2 ConTempo

We begin our discussion by first formulating the TRE task. Then, using a comparison of relational representations, we will demonstrate the difficulty that a neural model faces in temporal reasoning and understanding the meaning of temporal labels. In particular, there is a significant correlation between the mistakes a TRE model makes and its degree of inference over the available symmetric and antisymmetric properties. Next, we will present our ConTempo method in detail, which enhances a model's awareness of the meaning of each temporal relation and each principles of temporal reasoning. Finally, we will empirically demonstrate the effectiveness of ConTempo and compare it with previous methods in increasing temporal relation awareness.

2.1 TRE: Task Formulation

TRE can be formulated as a classification task. Given an input sequence $S = [t_1, \ldots, t_n]$ and two events represented by two distinct tokens from the sequence, $e_1, e_2 \in S$, where $e_1 \neq e_2$, the TRE model should classify the relation between the event pair (e_1, e_2) into one of several temporal relations, $\mathcal{R}$. Different corpora have different sets of temporal relations. Most TRE benchmark datasets aspire to embody Allen's (1981) 13 original temporal relations, but creating datasets with such fine-grained annotations is challenging. Attempts to utilize all 13 temporal relation types have failed due to low inter-annotator agreement. Therefore, recent TRE datasets typically simplify the relation set to a more coarse subset of the original set. As listed in Table 1, MATRES has 4 relation types, and TBD uses 6 relation types.

2.2 Temporal Relation Labels Awareness

After simplification, Allen's (1983) full temporal interval calculus can no longer be maintained. But interval symmetry and antisymmetry (as shown in Equation 1) still apply within both MATRES and TBD.[2] In our opinion, it is crucial for TRE models to reason with intervals in order to yield good evaluation results on benchmarks—if the models know A happened before B but do not know that B happened after A, can we really say that the model understands time in narratives?

$$\begin{aligned} A \text{ INCLUDES } B &\iff B \text{ IS\_INCLUDED } A \\ A \text{ BEFORE } B &\iff B \text{ AFTER } A \\ A \text{ EQUAL } B &\iff B \text{ EQUAL } A \end{aligned} \tag{1}$$

Symmetry and antisymmetry also provide us with an opportunity to test a model's understanding of relation labels. In particular, we can measure the similarity of the model's representation of a relation $\mathbf{h}_r^{\rightarrow}$ in comparison to its dual $\mathbf{h}_r^{\leftarrow}$:

$$\text{sim}(\mathbf{h}_r^{\rightarrow}, \mathbf{h}_r^{\leftarrow}) = \text{sim}(\mathbf{h}_{(e_1,e_2)}, \mathbf{h}_{(e_2,e_1)}), \tag{2}$$

It is relatively commonplace in the context of TRE to speak of a representation of a temporal relation, although this in fact refers to the representation of one of its instances over some event pair $\mathbf{h}_{(e_1,e_2)}$. Therefore, we will speak of the two interchangeably. If the relation between e_1 and e_2 is symmetric, then we can interpret high similarity as a sign of awareness concerning the relation's meaning because the relation and its inverse are nearly the same. On the other hand, if the relation is antisymmetric, we should expect the relations to be distinct and, therefore, we should interpret low similarity as a good sign. Here, we measure similarity with cosine similarity ($\text{sim}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$).

2.3 Temporal Relation Awareness of Fine-tuned BERT-based Models

In this section, we examine the extent to which typical, fine-tuned BERT-based models can be said to understand temporal relations and the temporal calculus. All contemporary TRE systems use this paradigm. Therefore, we experimented with a baseline model (described in Equation 3) to establish a foundation for the capacity of TRE models to summarize the temporal interval calculus.

[2]We ignore the VAGUE relation in this part of the analysis because whether symmetry applies to the VAGUE relation requires more careful consideration (Appendix D.3).

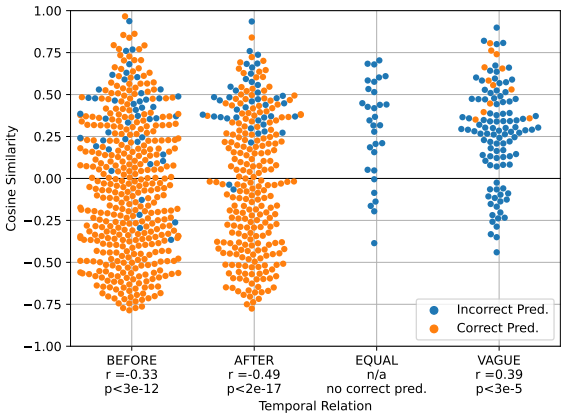

Figure 1: Similarity measures of event pair representations with their inverses. Each dot represents a relation in the MATRES test set: **green** means the model classified the relation correctly and **red** means the model made a mistake. The box plots underneath show the quantile statistics. There is an obvious negative correlation between the correctness of the prediction and the similarity measure. This is evident from observing that the incorrect predictions (red points) are located in the upper region of the figure for BEFORE and AFTER.

The baseline model has three components: a pretrained encoder (we use a RoBERTa-large model (Liu et al., 2019)), a 2-layer MLP module (f_1) that encodes the relation of the event pair under a unified representation scheme, and another single layer MLP (f_2) that decodes the relational representation into logits for each, different relational type:

$$\begin{aligned} \mathbf{E} &= \text{encoder}(S) = [\mathbf{h}_1, \ldots, \mathbf{h}_n], \\ \mathbf{h}_{(e_1,e_2)} &= f_1([\mathbf{h}_{e_1} \parallel \mathbf{h}_{e_2}]), \\ \mathbb{P}_{\mathcal{R}}(e_1, e_2) &= \text{softmax}(f_2(\mathbf{h}_{(e_1,e_2)})). \end{aligned} \tag{3}$$

The implementation details and hyperparameter settings are described in Appendix A.

Baseline Model Temporal Awareness Figure 1 shows the similarity between the relational representations $\mathbf{h}_r^{\rightarrow}$ and their symmetric duals $\mathbf{h}_r^{\leftarrow}$ as generated by the baseline model. We can see an obvious negative correlation between the correctness of the prediction and the similarity measure. This is evident from observing that the incorrect predictions (**red** points) are located in the upper region of the figure for BEFORE and AFTER. Furthermoe, the baseline model often assigns similar representations to relations and their symmetric duals, even when the two are diametrically opposite in meaning. As shown by the box plot, the first quantile of both the BEFORE and AFTER relations occupy

the upper part of the figure, which corresponds to high similarity. This shows that while fine-tuned models might have some basic perception of the significance of these temporal labels, it is by no means efficient, as all temporal labels are presented to the model as independent.

2.4 ConTempo: The Method

Contrastive learning builds on the idea of pulling similar (positive) data points (x, x^+) closer together and pushing dissimilar (negative) data points (x, x^-) apart by adding a contrastive loss term to the training objective (Robinson et al., 2021). We thus need to define positive and negative samples for temporal relation instances, i.e., event pairs. In this section, we will explain how positive and negative samples are constructed based on whether the relation is symmetric (EQUAL) or antisymmetric (BEFORE, AFTER, INCLUDES, and IS_INCLUDED), and define how we compute the temporally contrastive loss term ℓ_{tc}.

Our implementation of ConTempo is inspired by SimCSE's (Gao et al., 2021) use of contrastive learning on natural language inference (NLI) data. We likewise use a normalized temperature-scaled cross-entropy loss (NT-Xent; Chen et al., 2020).

Antisymmetric Relations For an antisymmetric relation $\mathbf{h}_r^{\rightarrow} = \mathbf{h}_{(e_1,e_2)}$, we regard the dual of the relation $\mathbf{h}_r^{\leftarrow} = \mathbf{h}_{(e_2,e_1)}$ as a hard negative sample. However, we cannot construct further positive samples using temporal reasoning. Instead, we follow SimCSE's unsupervised approach to obtain positive pairs through the use of independently sampled *dropout masks*. Standard transformers will apply dropout (default $p = 0.1$) at different locations in the model. Therefore, we simply feed the same input to our model *twice* to obtain two relational representations $\mathbf{h}_r^{\rightarrow}$ and $\mathbf{h}_r^{+}$ that are slightly different; different dropout masks having been applied during the two encoding passes. Therefore, we can compute the ConTempo loss by:

$$\ell_{\text{tc}} = -\log \frac{e^{\text{sim}(\mathbf{h}_r^{\rightarrow},\mathbf{h}_r^{+})/\tau}}{e^{\text{sim}(\mathbf{h}_r^{\rightarrow},\mathbf{h}_r^{+})/\tau} + e^{\text{sim}(\mathbf{h}_r^{\rightarrow},\mathbf{h}_r^{\leftarrow})/\tau}}, \quad (4)$$

where τ is a temperature hyperparameter.

Unlike SimCSE, however, we did not treat other relational representations in the same mini-batch as negatives. While we are certain that the relation yields the exact dual when we reverse the order of the two events ((e_1, e_2) *vs.* (e_2, e_1)), the comparison is less clear when it is made between different event pairs. Consider the following examples:

(1) Alan Turing **studied**$_{e_1}$ at Cambridge for his undergraduate degree and **attended**$_{e_2}$ Princeton for his Ph.D.

(2) I **failed**$_{e_3}$ my calculus course and then **cried**$_{e_4}$ all night in a McDonald's.

While both temporal relations $r_1 = (e_1, e_2)$ and $r_2 = (e_3, e_4)$ share the same BEFORE label, it is difficult to say whether they can be regarded as a positive pair, because they are different in many respects. First, we know that e_1 and e_2 are years apart but e_3 and e_4 happen on the same day. Second, we infer BEFORE for different reasons. We know that e_1 happened before e_2 through common-sense: a person needs to complete their undergraduate study before working towards a Ph.D. But we conclude that e_3 happened before e_4 because of *and then* and a likely causal relation between failing the course and crying. We want to preserve these differences and therefore avoid pushing the representations of r_1 and r_2 closer together.

We also do not need to consider relations with different labels, such as $r_1^{\rightarrow} = (e_1, e_2)$ and $r_2^{\leftarrow} = (e_4, e_3)$, as negative samples. Intuitively, contrastive learning is more effective when the negative samples occur nearby but should be far apart (Robinson et al., 2021). Since event pairs from different sentences contain drastically different syntactic and semantic information, it is not a serious problem for the classifier to distinguish them.

Symmetric Relations For a symmetric relation $\mathbf{h}_r^{\rightarrow} = \mathbf{h}_{(e_1,e_2)}$, we use the dual of the relation $\mathbf{h}_r^{\leftarrow} = \mathbf{h}_{(e_1,e_2)}$ as an additional, positive sample. Unlike with antisymmetric relations, we follow SimCSE and use other relations in the same mini-batch (B) as negative samples:

$$\ell_{\text{tc}} = -\log \frac{e^{\text{sim}(\mathbf{h}_r^{\rightarrow},\mathbf{h}_r^{\leftarrow})/\tau}}{\sum_{i \in B}\left(e^{\text{sim}(\mathbf{h}_r^{\rightarrow},\mathbf{h}_i)/\tau} + e^{\text{sim}(\mathbf{h}_r^{\leftarrow},\mathbf{h}_i)/\tau}\right)}. \quad (5)$$

We choose a different design here than for antisymmetric relations because it is impossible to construct *true* hard negative samples for symmetric relations. Also, we do not check whether other samples in the mini-batch have the same relation label as $\mathbf{h}_r^{\rightarrow}$ because EQUAL is the least common relation type. With only 3.8% in MATRES and 1.7% of the relations in TBD, it is very rare that there are multiple EQUAL relations in the same

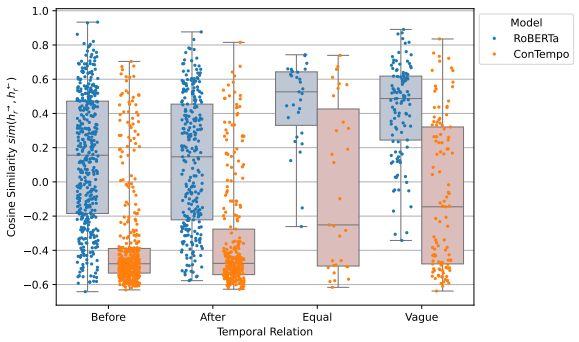

Figure 2: The efficacy of ConTempo at increasing the model's awareness of temporal relations and their implications through temporal reasoning.

mini-batch, so the effect of a colliding relation type is negligible.

Final Training Objective The ConTempo loss is added to the final training objective with a scaling hyperparameter γ (Equation 6). The classification loss ℓ_{clf} is the standard cross-entropy loss computed using the model's output logits:

$$\ell = \ell_{\text{clf}} + \gamma \cdot \ell_{\text{tc}}. \quad (6)$$

2.5 Experiment: ConTempo at Increasing Temporal Relation Awareness

Here we present the effectiveness of ConTempo at increasing temporal relation awareness. In this section, we trained an enhanced ConTempo baseline that is identical to the fine-tuned baseline but with ConTempo loss added to the model. We trained the model using the same hyperparemeters for the same amount of time.

In Figure 2, we compare ConTempo with baselines using the same similarity measures. We can observe that the similarity of the model's representation of a relation in comparison to its dual drops significantly after ConTempo is applied. Furthermore, now we can observe that the vast majority of similarity measures occupy the lower section of the figure, suggesting that ConTempo is effective at making antisymmetric relations generate representations that are distinct from their duals.

Unfortunately, we also see a similar drop in similarity for the symmetric relation (EQUAL). We believe this is caused by a data imbalance as there are too few EQUAL instances to balance the overall effect of the antisymmetric relations.

Method	MATRES	Antisym. Rel. Sim	Sym. Rel. Sim
RoBERTa	80.76	0.127	0.424
Data Expansion	80.16	-0.205	0.148
Relative Time	81.01	-0.237	0.144
Logical Constraints	81.26	-0.302	-0.102
ConTempo (Ours)	**82.01**	**-0.352**	-0.080

Table 2: The effectiveness of different methods at improving temporal relation awareness. The table reports each method's performance together with the average similarity measure for all the antisymmetric and symmetric relations.

2.6 Previous Attempts at Increasing Temporal Calculus Awareness

Does similarity reduction really lead to better temporal label awareness? Also, are there any other ways to increase temporal label awareness? Unfortunately, it is impossible to measure temporal label awareness directly, but we can infer it indirectly by observing the model's performance.

In this section we compared ConTempo with three previous methods that attempted to address similar problems. In the following paragraphs, we elucidate these methods, as they could also increase temporal relation awareness. In Table 2, we compare the effectiveness of all three with ConTempo. We use the same model setting for each method for a fair comparison. Overall, ConTempo is the most effective at reducing similarity. It also has the best performance on MATRES. This suggests ConTempo could help the model create more effective representations for TRE prediction.

Inverse Data Expansion Huang et al. (2023) simply expand the training set by adding all the inverse relations. This straightforward approach, however, does not address the issue that temporal relation labels are presented independently.

Relative Time Prediction Wen and Ji (2021) employed a relative timestamp prediction component to regulate the training process by predicting a real number ("relative event time," RET) that indicates the relative position of events in a timeline. This component is trained to maximize the distance between events in antisymmetric relations and to minimize the distance between events that are equal. This approach shares broad similarities with contrastive learning but faces two major challenges. First, relative event-time prediction in RET is ultimately computed for each single event individually, which is impossible. By contrast, ConTempo considers events in pairs and accounts only

for relative information. Second, this model cannot specify the absolute distance between events, meaning it does not differentiate between events that occur seconds apart and those that are days apart. It also makes predicting chains of events (e.g., $e_1 < e_2 < \ldots < e_5$ where $<$ indicates AFTER) difficult, as the model needs to balance various data points across the entire dataset and precisely position each event within the relative time range. ConTempo does not face this issue as the time distance between any two events (e_1, e_2) remains consistent regardless of the direction.

Logical Constraints Wang et al. (2020) proposed to impose soft logical constraints by adding a loss term that is calculated based on the model's output logits. In particular:

$$\ell_S = \sum |\log r(e_1, e_2) - \log \bar{r}(e_2, e_1)|, \quad (7)$$

where $r(e_1, e_2)$ denotes the logits the model outputs for relation type r given (e_1, e_2) as the input, and $\bar{r}$ denotes the dual of the relation. Other logical constraints such as transitivity are also enforced by Wang et al. (2020); we temporarily skip those as they are not relevant to the topic of the current section. We will return to a discussion of other types of constraint in Section 3.3.

Logical constraint loss has some surface-level resemblance to ConTempo loss, but their method focuses on the **output logits** whereas ours focuses on regulating the quality of the models' representations. The major motivation for Wang et al. (2020) to add logical constraints was to avoid label conflicts; enhancing the model's understanding of temporal relation labels was secondary.

3 The Unified Framework

Our survey of previous systems suggests that the reusability of innovations of TRE has been limited. For example, papers working on GNN-encoding of document-level information (Zhang et al., 2022; Mathur et al., 2021; Yao et al., 2022) rarely incorporate other published innovations, such as additional common-sense information, in their models. We call for more collaboration between different directions of TRE research.

In this section, we argue that ConTempo is compatible with various recent innovations across different research directions. In particular, ConTempo is complementary to: (1) GNN encodings of event and context information, (2) incorporating common-sense information, and (3) the soft enforcement of logical constraints.

3.1 Event and Context Encoder

All natural language understanding models need methods to encode their input sequence into vectorized representations, and to extract the relevant event and context representations. Currently, researchers typically use pre-trained language models such as BERT (Devlin et al., 2019) or RoBERTa to encode the input sequence because of the rich information these models gather during pre-training.

In TRE, an event is usually described by a complete phrase or even an entire clause. But since it is hard to annotate a discontinuous sequence, TRE corpora typically label the semantic head as the sole *event trigger* to represent the complete event. Extracting representations from only the event triggers may not obtain all the relevant information to make a correct prediction, however.

Therefore, TRE systems now utilize Graph Neural Networks (GNNs) to encode additional information about events and context. Zhang et al. (2022) proposed using dependency trees to help the model access different event arguments. TIMERS (Mathur et al., 2021) proposed using a GNN to encode three graphs: a sentence structure graph, a temporal structure graph, and a discourse structure graph. More recently, MulCo (Yao et al., 2022) reported better results using only the sentence ($\mathcal{G}_S$) and temporal ($\mathcal{G}_T$) structure graphs. They achieved better results with fewer graphs by using multi-scale contrastive learning between the fine-tuning of BERT and the training of the GNN to decrease GPU memory usage. Unlike ConTempo, their contrastive learning is not used to enhance the understanding of temporal labels.

We add MulCo to the unified framework by concatenating the GNNs' outputs to the event representations as $\tilde{\mathbf{h}_e} = [\mathbf{h}_e^{\mathcal{G}_S} \parallel \mathbf{h}_e^{\mathcal{G}_T} \parallel \mathbf{h}_e]$.

Please refer to Yao et al. (2022) for the detailed description of their model. We provide only a quick overview in Appendix B.

3.2 Common-sense Information

Similar to event and context encoding, we can also add additional features from external sources to enhance the representation. The feasibility of our unified framework is demonstrated by admitting temporal common-sense information.

Earlier TRE methods (Ning et al., 2018a, 2019; Wang et al., 2020) used temporal information de-

GPT-3 Prompt:

But the great-uncle of 6-year-old shipwreck survivor rafter Elian Gonzalez only ducked his head and walked faster.
Q: In about 10 words describe how long does the event "**ducked**" typically last.
A: **Ducking typically lasts only a few seconds.**

Figure 3: An example prompt and response for GPT-3. The context sentence is highlighted in **orange**, the event trigger, in **red** and GPT-3's response, in **green**.

rived from simple heuristics, or from corpora such as TempProb, which contains typical temporal-relation information between event triggers. Unfortunately, TempProb's coverage and accuracy is limited. Alternatively, the MC-TACO corpus (Zhou et al., 2019) was developed for temporal commonsense knowledge, presenting typical duration information through answers to multiple-choice questions. This approach, however, requires the model to select a typical duration from a restricted set of pre-defined options, and is incapable of addressing typical granularity and underspecificity problems. Our system is the first to apply temporal commonsense information from LLMs for TRE purposes.

We used GPT-3 (`text-davinci-003`), employing the template illustrated in Figure 3, to obtain descriptions of the typical durations of events in MATRES. We then use our system's underlying RoBERTa encoder to encode a description of the typical duration, and use the `[CLS]`-pooler output as the representation, denoted as $\mathbf{h}_e^{\text{dur}}$.[3]

$$\begin{aligned} \hat{\mathbf{h}_e} &= [\mathbf{h}_e^{\text{dur}} \parallel \tilde{\mathbf{h}_e}] \\ \mathbf{h}_{(e_1,e_2)} &= f_1([\hat{\mathbf{h}_{e_1}} \parallel \hat{\mathbf{h}_{e_2}}]) \end{aligned} \tag{8}$$

3.3 Training Objective

The third type of innovation is the modification of the training objective. ConTempo itself is an innovation to the training objective.

In this section, we show that ConTempo is compatible with other techniques that have been applied to training objectives. We have incorporated the aforementioned soft logical constraints (Wang et al., 2020) into the unified framework.

We add both the symmetric loss (ℓ_S) and the conjunction loss (ℓ_C) to the loss function. There is a bit of overlap between Wang et al.'s (2020) symmetric loss and ConTempo loss, but as previously discussed, we focus on regulating the representation whereas Wang et al.'s (2020) focus on avoiding label conflicts. ConTempo and the logical constraints complement each other, however, as ConTempo does not attempt to treat transitivity within contrastive learning. Instead, it is covered by conjunction restrictions. The loss terms are:

$$\begin{aligned} \ell_S &= \sum |\log r(e_1,e_2) - \log \bar{r}(e_2,e_1)|, \\ \ell_C &= \sum |L_{t_1}| + \sum |L_{t_2}| \end{aligned} \tag{9}$$

where L_{t_1} and L_{t_2} are defined as:

$$\begin{aligned} L_{t_1} &= \log r_1(e_1,e_2) + \log r_2(e_2,e_3) - \log r_3(e_1,e_3), \\ L_{t_2} &= \log r_1(e_1,e_2) + \log r_2(e_2,e_3) - \log(1 - r_4(e_1,e_3)). \end{aligned} \tag{10}$$

See Wang et al. (2020) for details, including the induction table for the conjunctive constraints. The final training objective is:

$$\ell = \ell_{\text{clf}} + \gamma \cdot \ell_{\text{tc}} + \lambda_S \cdot \ell_S + \lambda_C \cdot \ell_C. \tag{11}$$

4 Experiments

4.1 Experimental Setup

The hyperparameter settings and training details are explained in Appendix A.

Data We experimented with both MATRES (Ning et al., 2018b) and TBD (Cassidy et al., 2014), the two popular corpora that are most often used to benchmark TRE systems. Both are derived from the TimeBank Corpus (Pustejovsky et al., 2003), which is annotated using the TimeML standard (Pustejovsky et al., 2005). The VAGUE labels are computed into the F1s for both MATRES and TBD but are regarded as negative labels.

TBD is a dense re-annotation of the TimeBank corpus where every possible pair of events/times in a given window is forced to be annotated regardless of whether there exists a specified temporal relation. As a result, TBD has ample data samples (12715 relations from 39 articles) but a relatively low inter-annotator agreement, with Cohen's (1960) κ ranging between 0.56 and 0.64.

MATRES aimed at reducing noise in TBD's annotation. It introduced the concept of *orthogonal temporal axes* for hypothetical, negated, opined, and static events. In practice, these events are ignored during temporal relation annotation. They further reduced the number of temporal relations

[3]The prompts and the generated responses from GPT-3 will be made publicly accessible online.

Model	CleanMATRES	MATRES	TimeBank-Dense
ChatGPT 3.5	65.29	25.94	53.06
ChatGPT 3.5 (CoT)	66.97	35.54	57.14
ChatGPT 4	76.47	41.64	67.58
ChatGPT 4 (CoT)	80.76	56.70	65.14
Fine-tuned GPT 3.5	73.41	55.27	61.71
LSTM (Cheng and Miyao, 2017)	-	73.4	62.2
SCS-EERE* (Man et al., 2022)	91.25	80.98	65.92
Joint Constrained Learning (Wang et al., 2020)	-	78.8	-
Relative Time* (Wen and Ji, 2021)	89.61	81.63	66.81
MulCo* (Yao et al., 2022)	91.34	82.08	67.51
TIMERS (Mathur et al., 2021)	-	82.3	67.8
Endpoint Comparisons (Huang et al., 2023)	-	82.6	68.1
ConTempo Baseline	91.35	82.01	67.78
ConTempo + Unified Framework (Ours)	**92.66**	**83.17**	**68.56**

Table 3: F1 on CleanMATRES, MATRES and TBD. ConTempo achieved substantial improvements over strong baselines. Rows with results that we reproduced using the same evaluation settings as our model are labelled with *.

down to only four types. They also simplified the annotation scheme by asking the annotators to compare only the start times of events, as the end time of a durative event can often be ambiguous and cause confusion. During our survey of TRE baselines, we found that there are actually several versions of MATRES being used by different research labs (see Appendix C). We use the version that is available on the Ning et al.'s (2018b) GitHub page[4] and reproduced several key baseline systems' performances using our unified evaluation standards.

During the development of ConTempo, we discovered a large number of annotation errors in both MATRES and TBD. To further reduce the noise in the annotations, we re-examined and corrected the test set annotations of MATRES, creating the CleanMATRES test set. We will detail the process and describe the data in Section D.

Baseline Systems We compared ConTempo's performance with seven prior systems: LSTM (Cheng and Miyao, 2017), SCS-EERE (Man et al., 2022), Joint Constrained Learning (Wang et al., 2020), Relative Time (Wen and Ji, 2021), MulCo (Yao et al., 2022), TIMERS (Mathur et al., 2021) and Endpoint Comparisons (Huang et al., 2023). We reproduced the results of SCS-EERE and Relative Time using their publicly available code. Additionally, we reimplemented MulCo according to the specifications provided in the paper.

LLM Baselines We incorporated the performance of LLMs as an additional baseline alongside the aforementioned baseline systems. Specifically, we reproduced the results from Yuan et al.'s (2023) using both ChatGPT-3.5 and ChatGPT-4. Chain-of-thought (CoT) was deployed in the experiments. Additionally, we fine-tuned GPT-3.5 utilizing OpenAI's API using the suggested settings to perform the task.

[4] https://github.com/qiangning/MATRES

Model	CleanMATRES
ConTempo + Unified Framework	92.66
- ConTempo	91.30 (-1.36)
- GNN	91.65 (-1.01)
- GPT-3 Typical Duration	92.06 (-0.60)
- Logical Constraints	92.14 (-0.52)

Table 4: Ablation study results.

4.2 Experimental Results

Table 3 reports the performance of our ConTempo model compared to other baseline models on CleanMATRES, MATRES, and TBD. The results of our models and our reproductions are the average over three different runs with different random seeds.

Overall, ConTempo brings a substantial performance increase to the TRE model. Even the simple ConTempo baseline's performance is in the same ballpark as previous state-of-the-art systems. When enhanced by other innovations under the unified framework, our model outperformed the previous state-of-the-art models by 1.32%, 0.57%, and 0.46% for the three datasets, respectively.

4.3 Ablation Study

In order to evaluate the efficacy of the proposed components in the unified framework, we conducted an ablation study, wherein we removed one

component at a time and reran the experiment using the aforementioned hyperparameters. The results of this study are presented in Table 4. Among the four components, contrastive learning demonstrated the most substantial impact on performance, with its removal leading to a 1.36% decrease in the model's effectiveness. The GNN module also makes a substantial contribution to overall performance. Finally, both GPT-3 typical duration information and logical constraints exhibit moderate effects on the system's performance.

The ablation study result confirms our initial hypothesis: the primary challenge for these models has been making sense of relational labels.

5 Conclusion

In this paper, we propose a novel ConTempo method that utilizes contrastive learning on the symmetric and antisymmetric properties to enhance the model's awareness of temporal relations. Using ConTempo, we observe an apparent change in the representation of relations generated by the model. Particularly, we see a significant reduction in the similarity between relation representations $\mathbf{h}_r^{\rightarrow}$ in comparison to their dual $\mathbf{h}_r^{\leftarrow}$.

ConTempo and GNN encoding, temporal common-sense information, and soft logical constraints can work in synergy to create a unified system that has achieved state-of-the-art performance across various benchmark corpora. Our success demonstrates the flexibility of ConTempo, and we call for more collaboration between different directions of TRE research.

TRE is a rare and curious case where fine-tuned small LMs perform considerably better LLMs. Even with the help of groundbreaking techniques such as chain-of-thought (Wei et al., 2022), the best performance ChatGPT can achieve is 52.4% on MATRES (Yuan et al., 2023) and 41.0% on TBD (Huang et al., 2023) — remarkably lower than the fine-tuned small language models' performance. This anomaly presents tremendous opportunity for both TRE researchers and LLM researchers. TRE can be used as a hard task for the LLMs' to analyse their capabilities at reasoning and comprehension. In conclusion, exploring the integration of existing TRE innovations with LLMs and incorporating LLMs as a key component within TRE frameworks could significantly enhance performance and drive forward the advancement of both fields.

Acknowledgement

This study is funded by the Canadian Safety and Security Program (CSSP) from Defence Research and Development Canada (DRDC) awarded to the Public Health Agency of Canada (CSSP-2018-CP2334: Incorporating Advanced Data Analytics into a Health Intelligence Surveillance System). We thank the Global Public Health Intelligence Network (GPHIN) and Epidemic Intelligence from Open Sources (EIOS) teams for their support.

Limitations

We only evaluate ConTempo's performance using RoBERTa. Other language models, such as DeBERTa (He et al., 2021), T5 (Raffel et al., 2020) and DistilBERT (Sanh et al., 2020) may provide even better performance. We leave further exploration for future work.

Due to restricted dataset resources, our unified model's evaluation was conducted only using TB-Dense and MATRES. There are other TimeML-based and even non-TimeML-based corpora related to TRE. We leave more evaluation to future work and encourage the community to reproduce our results on more datasets.

References

James F. Allen. 1981. An interval-based representation of temporal knowledge. In *In Proceedings 7th IJCAI*, pages 221–226.

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.

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, Baltimore, Maryland. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607. PMLR.

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.

Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46.

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

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *International Conference on Learning Representations*.

Quzhe Huang, Yutong Hu, Shengqi Zhu, Yansong Feng, Chang Liu, and Dongyan Zhao. 2023. More than Classification: A Unified Framework for Event Temporal Relation Extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9631–9646, Toronto, Canada. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *ICLR 2019*.

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. 2021. TIMERS: Document-level Temporal Relation Extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 524–533, Online. Association for Computational Linguistics.

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, Haoruo Peng, and Dan Roth. 2018a. Improving Temporal Relation Extraction with a Globally Acquired Statistical Resource. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 841–851, New Orleans, Louisiana. 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, Victoria Ng, Erin Rees, Simon De Montigny, and Gerald Penn. 2023. Discourse Information for Document-Level Temporal Dependency Parsing. In *Proceedings of the 4th Workshop on Computational Approaches to Discourse (CODI 2023)*, pages 82–88, Toronto, Canada. Association for Computational Linguistics.

James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Rob Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, and Marcia Lazo. 2003. The TimeBank corpus. *Proceedings of Corpus Linguistics*.

James Pustejovsky, Bob Ingria, Roser Sauri, Jose Castano, Jessica Littman, Rob Gaizauskas, Andrea Setzer, Graham Katz, and Inderjeet Mani. 2005. The specification language TimeML. *The language of time: A reader*, pages 545–557.

James Pustejovsky, Kiyong Lee, Harry Bunt, and Laurent Romary. 2010. ISO-TimeML: An International Standard for Semantic Annotation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

James Pustejovsky, Marc Verhagen, Roser Sauri, Jessica Littman, Robert Gaizauskas, Graham Katz, Inderjeet Mani, Robert Knippen, and Andrea Setzer. 2006. TimeBank 1.2.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter.

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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

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.

Hao-Ren Yao, Luke Breitfeller, Aakanksha Naik, Chunxiao Zhou, and Carolyn Rose. 2022. Multi-Scale Contrastive Co-Training for Event Temporal Relation Extraction.

Jiarui Yao, Haoling Qiu, Bonan Min, and Nianwen Xue. 2020. Annotating Temporal Dependency Graphs via Crowdsourcing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5368–5380, Online. Association for Computational Linguistics.

Chenhan Yuan, Qianqian Xie, and Sophia Ananiadou. 2023. Zero-shot Temporal Relation Extraction with ChatGPT.

Shuaicheng Zhang, Qiang Ning, and Lifu Huang. 2022. Extracting Temporal Event Relation with Syntax-guided Graph Transformer. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 379–390, Seattle, United States. Association for Computational Linguistics.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "Going on a vacation" takes longer than "Going for a walk": A Study of Temporal Commonsense Understanding. 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 3363–3369, Hong Kong, China. Association for Computational Linguistics.

A Training Details

We use `roberta-large` as the pre-trained encoder. The best model is selected with the highest development set performance among 20 epochs of training, optimized using AdamW (Loshchilov and Hutter, 2019) with an initial learning rate of 1e-5 and a 0.01 weight decay. We use a linear scheduler with a warm-up ratio of 0.1 and a batch size of 64. The dropout rate is 0.1 across the system. For ConTempo, we determined $\tau = 0.05$, $\gamma = 1.0$ using a grid search over $\{0.1, 0.05, 0.01\} \times \{1, 0.1, 0.01\}$.

For the components in the unified systems, we follow the hyperparameter suggested by the corresponding papers (Yao et al., 2022; Wang et al., 2020).

B MulCo

MulCon utilized a sentence structure graph $\mathcal{G}_S$ and a temporal structure graph $\mathcal{G}_T$ to encode document-level information. The sentence structure graph has three types of nodes: the document, the sentences and the tokens. All sentences are connected to the document node and each token node is connected to the sentence it located in. For an event, $\mathcal{G}_S$'s embedding of the event trigger token is used.

The temporal structure graph, on the other hand, also has three types of nodes. The document creation time (DCT) node, the time expressions, and the events. Each time expression is connected to the DCT, and each event is connected to a time expression if there is an E2T link in the corpus. Finally, each event node has an edge point to itself.

C MATRES Versions

We found two versions of MATRES that has been used by different works. Ning et al. (2018b) released a version on GitHub [5] that contains 837 relations in the Platinum test set. Wang et al. (2020) released another version[6] with 818 relations in the Platinum test set. The difference (19 relations) could constitute as 2.32% of the test set data. We use Ning et al.'s (2018b).

D CleanMATRES

As we previously mentioned, we identified a large quantity of annotation errors within both MATRES and TBD. This is not surprising, however, as both corpora reported adequate inter-annotator agreements. TBD reported κ ranged from 0.56 to 0.64 and MATRES reported a good — but not perfect — κ of 0.84 (Table 6). In this section, we will first provide a description of the annotation errors

[5] https://github.com/qiangning/MATRES

[6] https://github.com/CogComp/JointConstrainedLearning/tree/main/MATRES

Relabelling	EQUAL		VAGUE		BEFORE		AFTER	
EQUAL	**10**	**32.26%**	15	13.27%	1	0.24%	0	0%
VAGUE	3	9.68%	**24**	**21.24%**	5	1.17%	5	1.86%
BEFORE	6	19.35%	39	34.51%	**405**	**95.52%**	0	0%
AFTER	10	32.26%	26	23.01%	4	0.94%	**256**	**95.17%**
Orth. Axes	2	6.45%	9	7.96%	9	2.12%	8	2.97%
Total	31	100%	113	100%	424	100%	269	100%

Table 5: Re-annotation of the MATRES test set. The statistics of the correct labels are highlighted in bold. Each column corresponds to an original MATRES label. Some relations involve events that should have been placed on one of the orthogonal axes.

Corpus	Cohen's κ
TimeBank-Dense	0.56-0.64
MATRES	0.84
MATRES EQUAL	1.0
MATRES VAGUE	0.75

Table 6: Detailed inter-annotator agreement of MATRES and TBD.

we identified in MATRES. Then, we will describe our re-annotation process. Finally, we will present an overview of the result of the re-annotation — CleanMATRES.

D.1 A Qualitative Analysis of the Annotation Errors

(3) He **won**$_{e_1}$ the Gusher Marathon, **finishing**$_{e_2}$ in 3:07:35.

Original Annotation: VAGUE
Correct Annotation: EQUAL

(4) The last surviving member of the team which first **conquered**$_{e_1}$ Everest in 1953 has **died**$_{e_2}$ in a Derbyshire nursing home.

Original Annotation: VAGUE
Correct Annotation: BEFORE

(5) More than 16,000 dead pigs have been **found**$_{e_1}$ **floating**$_{e_2}$ in rivers that provide drinking water to Shanghai.

Original Annotation: EQUAL
Correct Annotation: AFTER

Examples (3-5) show three examples of incorrect annotations in MATRES. In example 3, the relation between *won* and *finishing* is labelled as VAGUE, but, nonetheless, the two mentions refer to two aspects of the same punctual event. For example 4, it is clear that the member *died* a long time after they *conquered* Everest. For example 5, since MATRES annotators were only comparing the start-time of events, the pigs *floating* in rivers must took place before the event that they were *found*.

The two examples presented in this study illustrate the main sources of annotation errors commonly found in data sets. As posited by Ning et al. (2018b), the first example's error likely stems from the annotators' lack of comprehension regarding time granularity and event coreference. MATRES annotations are created by asking annotators two heuristic question:
(Q1) Is it possible that e^1_{start} is before e^2_{start}?
(Q2) Is it possible that e^2_{start} is before e^1_{start}?

The heuristic questions, Q1 and Q2, do not adequately address these concepts, necessitating the need to develop improved annotation guidelines. By effectively conveying the intricacies of time granularity and event coreference, the frequency of such errors may decrease, although perhaps with a lower κ. The second example's mistake should perhaps be attributed to a misunderstanding of the text.

D.2 CleanMATRES Annotation

We undertook a re-annotation of the MATRES test set. The test set were labelled independently by two authors of this paper, who were then forced to reconcile where they disagreed. Our pass through the test set reveals a significant number of annotation inaccuracies, particularly in the cases of the EQUAL and VAGUE relations. As detailed in Table 5, we found a mere 32.26% of the EQUAL and 21.24% of the VAGUE edges to have been accurately annotated.

D.3 Discussion: Vague Relations

We want to take the opportunity to discuss the issues related to the VAGUE relation. The VAGUE relation, which could be better called the *underspecified* relation, describes relations that are uncertain due to insufficient context. However, this certainty is highly subjective to the annotator and hard to rigorously quantify. TBD adopted an "80% rule" that instruct annotators to label a relation as vague if they are "80% confident that it was the writer's intent that a reader infer that relation." As a result, inter-annotator agreement on vague edges is the lowest among all relation types.

MATRES took a different approach and treats certain combinations of the start time comparisons as vague relations. The approach attenuates the issue but by no means solve the issue. The two heuristic questions are also subject and involves a degree of uncertainty. Furthermore, similar to EQUAL relations, a simple heuristic question mistake could result in incorrect VAGUE relation annotation.

With all the aforementioned issues, CleanMATRES excludes all VAGUE relations from the test set. We only consider BEFORE, AFTER, and EQUAL relations during the evaluation of CleanMATRES. We also did not include the analysis of VAGUE relations in Section 2, as it requires a more thorough examination of the application of temporal calculus when underspecificity is involved. Allen's (1983) original temporal calculus assumes all events are intervals with well-defined endpoints. Yao et al. (2020) and Niu et al. (2023) provided a more in-depth analysis to the VAGUE edges and underspecificity.