Will LLMs Replace the Encoder-Only Models in Temporal Relation Classification?

Gabriel Roccabruna, Massimo Rizzoli, Giuseppe Riccardi
Signals and Interactive Systems Lab
University of Trento, Italy
{gabriel.roccabruna, massimo.rizzoli, giuseppe.riccardi}@unitn.it

Abstract

The automatic detection of temporal relations among events has been mainly investigated with encoder-only models such as RoBERTa. Large Language Models (LLM) have recently shown promising performance in temporal reasoning tasks such as temporal question answering. Nevertheless, recent studies have tested the LLMs' performance in detecting temporal relations of closed-source models only, limiting the interpretability of those results. In this work, we investigate LLMs' performance and decision process in the Temporal Relation Classification task. First, we assess the performance of seven open and closed-sourced LLMs experimenting with in-context learning and lightweight fine-tuning approaches. Results show that LLMs with in-context learning significantly underperform smaller encoder-only models based on RoBERTa. Then, we delve into the possible reasons for this gap by applying explainable methods. The outcome suggests a limitation of LLMs in this task due to their autoregressive nature, which causes them to focus only on the last part of the sequence. Additionally, we evaluate the word embeddings of these two models to better understand their pre-training differences. The code and the fine-tuned models can be found respectively on GitHub[1].

1 Introduction

An important ability in understanding information flows such as news is to recognize the temporal relations of events, which happened, are happening, or will happen, to order them into a coherent storyline. Indeed, temporal relations are utilized in many natural language processing tasks such as narrative understanding (Song and Cohen, 1988; Mousavi et al., 2023), story generation (Han et al., 2022), summarization (Liu et al., 2009; Gung and Kalita, 2012) and temporal question answering (Shang et al., 2022; Kannen et al., 2023).

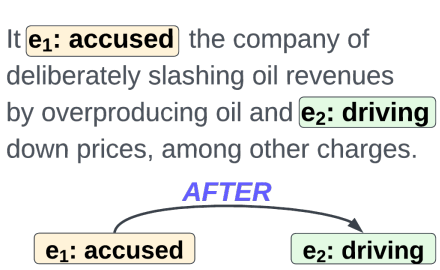

Figure 1: An example taken from MATRES corpus for the Temporal Relation Classification task, in which the accusation event follows the driving event. The relation between the two event triggers, namely e_1:*accused* and e_2:*driving*, is annotated with a directed arc and the label *AFTER*.

The automatic recognition of temporal relations (e.g. *before* or *after*) is referred to as Temporal Relation Classification (TRC). Figure 1 depicts an example of the TRC task, as defined in the TempEval challenges (Verhagen et al., 2007; Pustejovsky and Verhagen, 2009; UzZaman et al., 2013), that is to predict the temporal relation *after* between the two given connected events, $e_1 : accused$ and $e_2 : driving$. The temporal relations used to annotate the corpora for training and testing models have been originally defined in Allen's interval algebra (Allen, 1983, 1984). In this, events are described as intervals rather than time points to handle explicit and implicit or vague time references.

In recent years, several works on the TRC task have focused on exploiting a variety of features to best represent the surrounding context of the two events. Some of these are syntactic (Zhang et al., 2022), semantic (e.g., event arguments) (Zhou et al., 2022) and discourse (Mathur et al., 2021) features. To utilize those features, models have been based on complex architectures constructed on top of encoder-only pre-trained language models

[1] https://github.com/BrownFortress/LLMs-TRC

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 20402–20415
November 12-16, 2024

such as BERT (Kenton and Toutanova, 2019) and RoBERTa (Liu et al., 2019). More recently, Large Language Models (LLMs) have become widely used in many natural language processing tasks, achieving state-of-the-art performance in sentiment analysis (Zhang et al., 2023), named entity recognition (Wang et al., 2023) and natural language inference (Chowdhery et al., 2023). Although LLMs have been evaluated on the temporal question answering task (Wei et al., 2023; Gupta et al., 2023; Dhingra et al., 2022), in which temporal relations are implicitly used, limited studies have been conducted on the performance of LLMs on the TRC task (Li et al., 2023; Yuan et al., 2023; Chan et al., 2023) experimenting with closed-sourced LLMs, limiting interpretability studies.

In this paper, we study the performance and the decision process of seven open and closed-sourced Large Language Models (LLM) in performing the task of Temporal Relation Classification (TRC) over three different publicly available benchmark corpora. Along with an example-label in-context learning (ICL) approach (Brown et al., 2020), we cast the TRC task into a Question Answering task form to create QA prompts. Furthermore, we use the Low-Rank Adaptation (LoRA) (Hu et al., 2021) technique to fine-tune Llama2 7B and Llama2 13B to measure the upper bound performance of such models. The results show that although the autoregressive LLMs with QA prompts perform better than example-label prompts, they struggle with the TRC task compared to smaller encoder-only models based on RoBERTa in all settings. We further investigate the possible reasons for this by analyzing the most contributing tokens to the prediction extracted with KernelShap (Lundberg and Lee, 2017) algorithm as an XAI attribute scorer. This analysis shows that LLMs tend to focus more on the last tokens of the target sentence due to their autoregressive nature, rather than using the entire context as encoder-only models do. We evaluate the word embeddings extracted from LLMs and RoBERTa to highlight the pre-training differences between these two models. We observe that RoBERTa still performs better than LLMs, suggesting that the gap between these two models is probably due to their different pre-training strategies.

The contributions of the paper are the following:

- Evaluation of seven LLMs including open and closed-sourced models with different parameter sizes and with ICL and LoRa approaches;
- Explainability studies on LLMs and RoBERTa models to understand the differences between the two models in their decision processes;
- Word embeddings evaluation and comparison between LLMs and RoBERTa model;

The remainder of the paper is organized as follows. Section 2 reviews the related works. In section 3, we formally define the TRC task. Section 4 presents the encoder-only model based on RoBERTa and the prompts for LLMs. In section 6, we present the results of the tested models and the explainability studies. In section 7, we present and discuss the error analysis. Finally, we present our conclusions.

2 Related works

Temporal Relation Classification task Temporal Relation Classification (or Extraction) models have predominantly used encoder-based pre-trained language models, such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), as a backbone. In particular, some studies have employed graph neural networks initialised with BERT and RoBERTa embeddings to model the semantic and syntactic context surrounding the events (Mathur et al., 2021; Zhou et al., 2022; Zhang et al., 2022). More recently, Cohen and Bar (2023) have been fine-tuned a RoBERTa model for answering interval relation reasoning questions to predict a given temporal relation class. Instead, LLMs have been tasked to answer multiple choice questions (Chan et al., 2023), which challenges them to understand the semantics of the different temporal relations classes.

Temporal QA The purpose of extracting temporal relations among events proposed in TimeML (Pustejovsky et al., 2003a) is to improve the performance of temporal Question Answering task (Llorens et al., 2015; Meng et al., 2017; Yang et al., 2023), which is to answer temporal grounded queries such as *"Which is the current US president?"*. Recently, the temporal QA task has been used for testing and challenging the temporal reasoning ability of LLMs (Wei et al., 2023; Tan et al., 2023) by querying the LLMs parametric knowledge with time-grounded questions.

Temporal Relations Corpora One of the first and largest corpora annotated with temporal relations is the TimeBank (Pustejovsky et al., 2003b) corpus. This corpus has been annotated using

the Time-ML scheme (Pustejovsky et al., 2003a). This scheme provides a definition used to identify events and defines a set of thirteen temporal relations which follow in principle the thirteen interval relations of Allen's interval algebra (Allen, 1983, 1984). Adaptations and refinements of the ISO Time-ML (Pustejovsky et al., 2010) scheme, mainly regarding the event definition and the number of relations, have been used to annotate most of the available corpora such as in Thyme-TimeML (Styler IV et al., 2014), TimeBank-Dense (Cassidy et al., 2014), RED (O'Gorman et al., 2016), MATRES, MAVEN-ERE (Wang et al., 2022) and TIMELINE.

3 Task Definition

The Temporal Relation Classification (TRC) task can be defined as follows. The corpus comprises a set of documents $\mathcal{D}$. A document $d \in \mathcal{D}$ is defined as a sequence of sentences $d = [s_1, ..., s_n]$, where a sentence is a sequence of tokens i.e. $s = [w_1, ..., w_n]$. A sentence is delimited by a full stop, exclamation or question mark. Each document in the corpus contains a set of annotated event triggers $E = \{e_1, ..., e_n\}$ where e is a span of tokens of a sentence of a document i.e. $e = [w_i, ..., w_j] \in s \; with \; i > 0, \; j \leq |s| \; and \; s \in d$. The TRC task is to assign a temporal relation r from a predefined set $\mathcal{R}$ to a given pair of connected events (e_i, e_j), where $e_i \neq e_j$. The set of relations $\mathcal{R}$ changes depending on the annotation scheme of the corpus as described in Section 5.1. Besides, the temporal relations $r \in (e_i, e_j)$ and $r^{'} \in (e_j, e_i)$ are always the opposite (e.g. *before* and *after*) i.e. $r = \neg r^{'}$ except when r is the relation *equal*, where by definition $r = r^{'}$. Indeed, in Allen's interval algebra, the 13 temporal relations are composed of *equal* plus six temporal relations and their corresponding opposites.

4 Models

In this section, we describe the encoder-only model and LLMs with fine-tuning and prompting approaches tasked with Temporal Relation Classification (TRC).

4.1 Encoder-only Architecture

In this work, we use RoBERTa (Liu et al., 2019) as an encoder-only model. This model has been pre-trained on the Masked Language Modelling (MLM) (Devlin et al., 2019) task. In this, the model is tasked to predict a masked token attending to the rest of the sequence.

Inspired by (Zhou et al., 2022), we have used the following architecture to put RoBERTa in place. The input of the models is the corresponding sentences containing the event pairs (e_i, e_j). The events can be in the same (intra-sentence) or in two different (inter-sentence) sentences. Formally, the input for intra-sentence events is $C = s_k, \; s_k \in d$ where $e_i, e_j \in s_k$ and for inter-sentence events is $C = s_i \oplus s_j, \; s_i, s_j \in d$ where $e_i \in s_i, e_j \in s_j$. For the latter, we concatenated the two sentences with a white space. The input C is fed into a pre-trained model to compute the word embeddings of input tokens. From these, we retrieve the embedding corresponding to the tokens of the two events. Then, the event embedding is created by aggregating all the relative sub-tokens generated by the byte pair encoding tokenizer (Sennrich et al., 2016) using the max pooling function. Aggregating all tokens is important since the verb tense is a relevant aspect of this task; therefore, the *-ed* sub-token (i.e. the last sub-token) can be an important feature for the event embedding. The two events embedding are then concatenated. Finally, the resulting concatenated vector is fed into a feed-forward linear layer followed by a softmax to make the prediction.

4.2 In-Context Learning and Fine-Tuning

Temp. Rel.	Questions
Before	Does e_i happen ***before*** e_j?
After	Does e_i happen ***after*** e_j?
Equal	Does e_i happen ***at the same time as*** e_j?
Includes	Does e_i ***temporally include*** e_j?
Is Included	Is e_i ***temporally included in*** e_j?

Table 1: Casting of TRC task to QA task. The two events are identified as e_i and e_j. In the actual prompt used for LLMs, we surrounded the two events with the tags `[event1][/event1]` for e_i and `[event2][/event2]` for e_j.

Large Language Models (LLM) have been pre-trained on a large scale of data using the autoregressive language model (Roth, 2000; Brown et al., 2020) as the objective task. Differently from MLM, the model has to predict the next token t_{k+1} using the previous tokens, i.e., the context sequence $t_0, ..., t_k$ only.

To evaluate LLMs, we experiment with in-

context learning (ICL) and fine-tuning approaches (Brown et al., 2020). In the ICL experiments, we evaluate the ability of the model to understand and, thus, tackle the task by using the pre-trained knowledge only. Moreover, by updating this knowledge with fine-tuning, we measure the upper-bound performance of such models.

Inspired by (Brown et al., 2020), we have translated the TRC task into a text-to-text task using the widely used example-label pattern, henceforth referred to as ***P***. In particular, the example in ***P*** is composed of the context C, i.e. the concatenation of the sentences containing the events as for the RoBERTa-based model, with the two events highlighted using two tags ("`[event1]`e_i`[/event1]`" and "`[event2]`e_j`[/event2]`"). This is followed by the symbol "->" and the target label. Thus, given the context, the model has to generate one of the temporal relation labels, i.e. $r \in \mathcal{R}$.

We have translated the TRC task into a question answering task to further investigate LLMs' reasoning capabilities. Motivated by LLMs' training on massive amounts of web-based data, we have designed questions that can be answered without prior knowledge about the temporal relation theory and/or formal annotation guidelines. Indeed, previous works have designed questions by involving interval reasoning (Li et al., 2023; Cohen and Bar, 2023). For instance, in those to identify the *before* relation the two following questions are asked "*Does event* e_1 *start before* e_2*?*" and "*Does event* e_1 *end before* e_2*?*". In this work, we have formulated one question for each temporal relation class i.e. *before, after, equal, includes* and *is included*. These questions are listed in Table 1. To let the model answer the question, we use the same context C of the prompt ***P***. Moreover, we have experimented with asking the model one question at a time ***QA₁*** and all the questions in a sequence ***QA₂***. In ***QA₂***, the model can use the generated response as additional context to answer the remaining questions.

We used the Low-Rank Adaptation (LoRA) technique (Hu et al., 2021) to fine-tune the LLMs, achieving an upper bound in the performance. LoRA is an efficient fine-tuning approach because it adds and trains only a small set of parameters (i.e. less than 1% of all parameters) to the model.

5 Experimental settings

5.1 Datasets

We have tested our models on TimeBank(TB)-Dense (Cassidy et al., 2014) and MATRES (Ning et al., 2018), which are widely used benchmarks, and TIMELINE (Alsayyahi and Batista-Navarro, 2023), which is a newly released corpus. TB-Dense is composed of 36 news articles, a subset of the TimeBank corpus, published in 1990 and 1998. TB-Dense has been annotated with 6 temporal relations i.e. *before*, *after*, *includes*, *is included*, *simultaneous*, and *vague*. MATRES includes all the 275 news articles used in the TempEval-3 challenge. All the news articles in the train and validation sets were written and published in the time range between 1990 and 2000, while in the test set, they are all dated 2013. The corpus has been annotated with four temporal relations *before*, *after*, *equal* and *vague*. TIMELINE is composed of 48 news articles published between 2020 and 2021 and has adopted the same temporal relation scheme of MATRES. We have used official train, development and test sizes and the label distributions are shown in Appendix A.

5.2 Evaluation metrics

All the models are evaluated using the micro-f1 score. Following the decision made for TIMELINE (Alsayyahi and Batista-Navarro, 2023), we have completely removed from MATRES and TB-Dense the *vague* class. This is because we want to focus only on temporal relations. The class *vague* is not a temporal relation (Wen and Ji, 2021; Zhou et al., 2022) as it has been used to handle ambiguities and disagreement during the annotation process (a.k.a. catch-all class). Indeed, we have used the class *vague* to map the examples for which the LLMs output gibberish or produce contradictory responses.

In TB-Dense in the prompts ***QA₁*** and ***QA₂*** for the label *simultaneous*, we have used the same question for the temporal relation *equal* as they have the same meaning.

6 Evaluation and Results

We have experimented with seven Large Language Models (LLM): five open-source, namely Llama2 7B, Llama2 13B, Llama2 70B (Touvron et al., 2023), Mistral 7B (Jiang et al., 2023), Mixtral 8×7B (Jiang et al., 2024), and two closed-sourced,

Models	MATRES			TIMELINE			TB-Dense		
	P	***QA₁***	***QA₂***	***P***	***QA₁***	***QA₂***	***P***	***QA₁***	***QA₂***
Mistral 7B	30.0	14.8	52.9	28.7	8.1	39.9	5.0	0.4	0.0
Mixtral 8×7B	27.7	28.1	58.0	36.1	30.2	53.2	8.5	12.3	13.1
Llama2 7B	31.2	14.8	56.3	41.8	9.7	58.1	21.7	1.6	0.6
Llama2 13B	36.7	8.5	31.1	41.8	8.0	28.3	27.9	3.3	24.3
Llama2 70B	36.6	37.0	**65.3**	39.4	48.0	**62.5**	27.1	9.3	**31.4**
GPT-3	54.0	8.0	55.6	7.0	20.3	57.3	2.7	2.5	0.5
GPT-3.5	41.2	29.6	61.2	11.7	12.2	58.5	19.0	24.6	12
Llama2 $7B_{\text{Fine-tuned}}$	71.4	77.2	82.0	57.2	**76.9**	55.9	45.0	4.7	49.3
Llama2 $13B_{\text{Fine-tuned}}$	76.5	81.6	**84.3**	61.3	30.5	41.5	**55.4**	3.7	48.7
RoBERTa	87.6			87.9			83.1		

Table 2: Results achieved by LLMs with in-context learning (ICL) and fine-tuning on MATRES (Ning et al., 2018), TB-Dense (Cassidy et al., 2014) and TIMELINE (Alsayyahi and Batista-Navarro, 2023) corpora. ***P*** refers to the example-label prompt. In ***QA₁*** and ***QA₂*** the TRC task is cast into two QA prompts. In ***QA₁*** the model answers one question at a time, while in ***QA₂*** the model uses as context its responses by answering the question in sequence. The results in bold are the best achieved among LLMs with ICL and fine-tuning.

namely GPT-3 (Brown et al., 2020) and GPT-3.5[2] (OpenAI, 2023). As described in section 4, we have adopted in-context learning (ICL) and fine-tuning approaches using the following prompts:

- ***P***: given two events and the corresponding context, i.e. the sentences including the events, the model generates the label (e.g., *before* or *after*) that identifies a temporal relation.
- ***QA₁***: given two events, the corresponding context, and a question for each class (shown in Table 1), the model answers one question at a time with *yes* or *no*.
- ***QA₂***: given the same context as in QA_1, the model answers all questions in sequence. In this setting, the generated responses become part of the context used to answer the following question.

Regarding the ICL experiments, we have experimented with zero and different numbers of few-shot examples on Llama2 7B. We have observed that the models achieve the best performance by using one example for each class of the corpus, i.e. five for TB-Dense and three for both MATRES and TIMELINE. Furthermore, to measure the impact on the performance of the few-shot example selection, we have sampled and frozen five sets of few-shot examples to create the context for all three prompt types.

The results [3] of ICL and fine-tuning experiments are reported in Table 2. The prompt ***P*** is effective for GPT-3 and GPT-3.5 on MATRES only. Moreover, while on MATRES and TIMELINE overall, the models achieve the worst results with ***QA₁***, the best results are achieved using the prompt ***QA₂***. One reason for this is that in the 12% of event-pair predictions, on average, the models with ***QA₁*** generate contradictory responses, i.e. answering *yes* to more than one question. Adding the generated answers to the context for the next question, i.e. the prompt ***QA₂***, zeroes the contradictory responses. Furthermore, Llama2 70B outperforms all the other models with ***QA₂*** prompt on all corpora. Besides, all the LLMs struggle with TB-Dense, probably due to a higher number of classes to predict.

Regarding the performance of individual LLMs with ICL, Llama2 70B outperforms all other open and closed-source models on all corpora. Furthermore, despite the 175B (billions) parameters, GPT-3 yields worse results compared with Llama2 7B on all corpora and Mixtral 8×7B on MATRES whose numbers of parameters are 7B and 12B[4] re-

[2]GPT-3 is davinci-002 and GPT-3.5 is gpt-3.5-turbo-0125

[3]ICL results have been averaged over the five prompts; fine-tuning results have been averaged over five runs.

[4]Mixtral 8×7B has 58 billion of parameters but at infer-

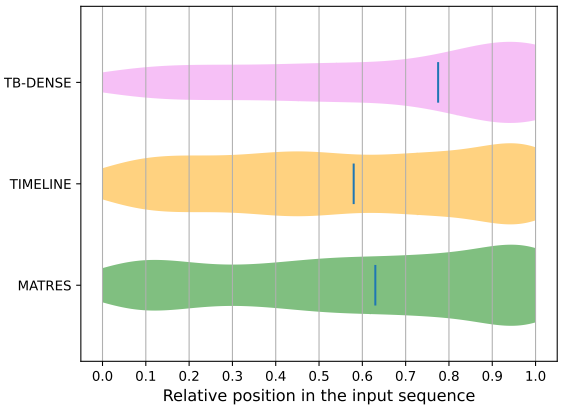

Figure 2: Distribution of the five tokens for each input sequence with the highest attribute score computed with *Llama2 7B* (Touvron et al., 2023) based on the input sequence. Corpora on the y-axis and relative position in the input sequence on the x-axis. The blue line is the median.

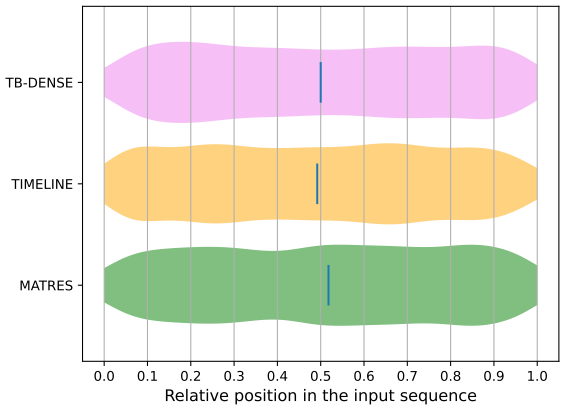

Figure 3: Distribution of the five tokens for each input sequence with the highest attribute score computed with *RoBERTa* (Liu et al., 2019) based on the input sequence. Corpora on the y-axis and relative position in the input sequence on the x-axis. The blue line is the median.

spectively. Furthermore, Llama2 7B outperforms Mistral 7B on all corpora and Mixtral 8×7B on TIMELINE and TB-Dense. Besides, LLama2 13B underperforms LLama2 7B in all corpora but TB-Dense.

To estimate the upper bound performance of LLMs, we have fine-tuned Llama2 7B and 13B on the three corpora using the same prompts of ICL but with a zero-shot approach. On MATRES, Llama2 13B fine-tuned using prompt $\boldsymbol{QA_2}$ achieves close results to the encoder-only model based on RoBERTa . Instead, Llama2 7B fine-tuned with $\boldsymbol{QA_1}$ scores the highest micro-F1 among LLMs on TIMELINE but is 11.0% inferior to the RoBERTa-based model. On TB-Dense, while achieving the best score compared with prompt ***P***, Llama2 13B scores the highest gap of 27.7% w.r.t. RoBERTa-based model. A possible reason for this is that TB-Dense has two additional temporal relations compared to MATRES and TIMELINE, making the training and inference more challenging.

Overall, the results achieved by LLMs in all settings are always outperformed by the encoder-only model based on RoBERTa. Indeed, RoBERTa scores improvements w.r.t ICL best models of 22.3% on MATRES, 25.4% on TIMELINE and 51.7% on TB-Dense. Although fine-tuning Llama2 7B and 13B substantially reduces this gap on MATRES, the differences are still high in the other two datasets.

6.1 Explainability studies

We have studied the gap in performance between LLMs and the encoder-only RoBERTa-based model using an attribution method called KernelShap (Lundberg and Lee, 2017). KernelShap is an additive feature attribution method based on Linear LIME (Mishra et al., 2017) and SHAP values (Lundberg and Lee, 2017), which gives a score to each input vector element based on its importance to the prediction.

From each input sequence in three test sets, we have computed [5] and extracted the five tokens with the highest attribution score. Regarding Llama2 7B with prompt ***P***, we have observed that 70% of these tokens are positioned in the few-shot context, while the remaining come from the target sequence (i.e. the context of the two events to make a prediction). To be comparable with RoBERTa model, we present the distribution of the tokens based on their position coming from the target sequence only. To do this, we have computed the relative positions, i.e. scaling into a 0 to 1 range, dividing them by the sequence length.

Figure 2 and Figure 3 show the violin plots of the distribution of the tokens with the highest attribute score based on their positions computed using Llama2 7B and RoBERTa respectively. Regarding Llama2 7B, most of the tokens with the highest attribution score are at the end of the se-

ence time it automatically selects and utilizes a subset of 12 billion.

[5] For this, we have used Captum library.

Models	Frozen Encoder			Full Fine-Tuning		
	MATR.	TIMEL.	TB-D.	MATR.	TIMEL.	TB-D.
Llama2 7B	75.2	64.8	68.0	79.4	64.9	77.3
Llama2 13B	76.6	66.6	68.9	82.8	69.8	77.7
Llama2 70B	75.9	**69.1**	65.7	81.5	67.2	72.4
RoBERTa	**80.5**	65.7	**71.4**	**87.6**	**87.9**	**83.1**

Table 3: Performance comparison between RoBERTa-based model (last row), and Llama2 7B, 13B, and 70B. *Frozen Encoder* reports the micro-F1 scores in percentage achieved by training the classification layer only. *Full Fine-Tuning* reports the results attained by fine-tuning also the encoder model. (`MATR.=MATRES`, `TIMEL.=TIMELINE`,`TB-D.=TB-Dense`)

quence, meaning that the model tends to use only the last few tokens to make a prediction. Conversely, the distribution of the encoder-only model is more uniform, meaning that the decision process of RoBERTa considers the entire sequence. This suggests that one of the reasons behind the gap in the performance between these two kinds of models is due to the different pre-training tasks,i.e., the masked language model task (Devlin et al., 2019) for Roberta and the autoregressive language model task for the LLMs.

6.2 Word Embedding analysis

We have compared the performance of the word embeddings generated by LLMs and RoBERTa models in the TRC task to investigate the differences due to the pre-training strategies.

To do this, we have used the architecture presented in Section 4 and replaced the encoder, i.e., RoBERTa, with Llama2 7B, 13B and 70B. We have experimented with training only the classification layer, i.e. freezing the weights of the encoder, and with full fine-tuning, using LoRa for the LLMs, to attain the upper-bound performance.

The results of these experiments are presented in Table 3. The micro-F1 scores attained with LLMs by training the classification layer only are higher than those achieved with the same models with ICL. However, RoBERTa still achieves the highest performance on MATRES and TB-Dense. Interestingly, LLama2 70B outperforms RoBERTa on TIMELINE. The possible reason for this is that TIMELINE contains many news articles related to the recent COVID-19 pandemic. Indeed, in more than 30% of target sequences in the training and test sets, there is one of the following words *covid-19*, *coronavirus*, *pandemic* and *vaccine*. Considering that RoBERTa was pre-trained in 2019, those tokens are out-of-vocabulary tokens for the model.

By further training the encoder, the models generally increase the performance on all corpora with an average improvement of 4% and 9% results on MATRES and TB-Dense respectively. Conversely, on TIMELINE, the improvement of LLama2 7B and 13B is considerably contained, and we observe a worsening in the performance of Llama2 70B. While RoBERTa gains an increment of 22.2%, providing additional evidence of the initial high presence of OOV tokens for the RoBERTa model.

Overall, the results suggest that the word embeddings yielded by RoBERTa are more effective in the TRC task, supporting the outcome of the explainability studies for which one of the probable reasons for the performance gap is in the different pre-training tasks.

7 Error Analysis

We have analyzed the error of the RoBERTa-based model, Llama2 70B and Llama2 13B fine-tuned by comparing the performance between intra and inter-sentences on MATRES as reported in Table 5. The encoder-only RoBERTa-based model achieves the highest intra- and inter-sentence performance. Besides, all three models underperform on the intra-sentences, where the highest difference is measured on Llama2 70B.

To investigate whether there is a subset of the test set with challenging examples for all the models, we have computed the intersection between the errors of the RoBERTa-based model and the correct predictions of Llama2 13B fine-tuned and Llama2 70B with ICL. The sizes of these intersections are 23.3% for Llama2 13B fine-tuned and 33.7% for Llama2 70B, which account for 2.8% and 4.0% of the test set, respectively. By manually

Models	MATRES			TIMELINE			TB-Dense				
	Bef	Aft	Eq	Bef	Aft	Eq	Bef	Aft	Eq	Incl	Is_Incl
Llama2 7B	71.9	6.4	2.1	70.1	25.7	0.0	0.2	30.1	0.3	9.5	0.0
Llama2 13B	2.0	53.5	0.7	5.0	57.7	2.5	10.6	41.9	3.3	11.0	5.1
Llama2 $7B_{\text{Fine-tuned}}$	87.2	77.9	0.0	81.9	81.2	0.0	63.7	34.6	0.0	0.0	0.0
Llama2 $13B_{\text{Fine-tuned}}$	88.6	82.1	0.0	68.3	51.8	0.0	65.8	51.9	0.0	0.0	0.0
RoBERTa	91.8	86.2	0.0	89.8	87.4	8.9	88.6	86.7	0.0	56.6	59.2

Table 4: Comparison of F1-scores in percentage for each class of each dataset achieved using the best prompt settings. (`Bef=Before, Aft=After, Eq=Equal, Incl=Includes, Is_Incl=Is Included`)

Models	Intra-sent.	Inter-sent.
RoBERTa	85.8	89.6
Llama2 $13B_{\text{Fine-tuned}}$	82.3	85.5
Llama2 70B	59.9	69.2

Table 5: Performance (micro-f1) of intra and inter-sentence event pairs on MATRES. Intra-sentence the event pairs are in the same sentence, while in inter-sentence they are in two different sentences. The distribution of intra and inter-sentences are 39.0% and 61.0% respectively.

inspecting these subsets, we have found that the errors are mainly due to misunderstanding of verb tenses such as past perfect continuous and future. While in the remaining errors of RoBERTa, we observed that there are challenging examples also for humans, as they require additional common sense knowledge and strong reasoning capabilities such as simulation reasoning (Tamari et al., 2020). Some examples of such cases are shown in Table 10 in Appendix A.

Regarding the impact of the selection of few-shot examples in ICL on LLMs, we observe that the standard deviation mainly depends on the type of prompt and the model as presented in Table 8 in Appendix A. Notably, the model and the prompt with the lowest performance variability on average are Llama2 70B with 2.2% and $\boldsymbol{QA_2}$ with 2.7%. In comparison, the few-shot selection has the highest impact on $\boldsymbol{QA_1}$ with 5.8% and Mistral 7B with 6.1%. Thus, although requiring a relevant amount of time and resources, tuning the few-shot samples on the development set might boost the performance of some models and prompts.

To better understand the performance of the models with ICL and fine-tuning, in Table 4 we report the F1-scores for each relation of the models using the best prompt settings[6]. We can observe that LLMs with ICL are biased towards a specific class, i.e. the class *after* for LLama2 13B and *before* for 7B. This is reduced with fine-tuning. Furthermore, the class *equal* is mispredicted by all the models on all three corpora, RoBERTa included. A possible reason is that *equal* is always the least frequent class counting for 2% to 4% of the total number of examples.

8 Conclusions

In this work, we have evaluated seven open and closed-sourced LLMs on the Temporal Relation Classification task with an in-context learning and fine-tuning approach. We have shown that the encoder-only RoBERTa-based model achieves the highest results compared to LLMs. Explainable studies suggest that one of the reasons for this gap is due to the different pre-training tasks. Finally, considering the low performance and the huge amount of computational resources needed at fine-tuning and inference time, LLMs might not be the best option for the TRC task compared to a more accurate and low-resource demanding RoBERTa-based model.

Future work A possible future work is to further investigate the pre-training task differences, by pre-training two models on the autoregressive and masked language model tasks using the same parameter size and training set. Another possible direction is to study a hybrid architecture to join the best performance of the RoBERTa-based models and the Large Language models such as Llama2 13B fine-tuned and Llama2 70B.

[6]The F1 scores for all the models can be found in Table 11 in Appendix A.

Limitations

We could not experiment with the largest open-source model due to limited resources. Furthermore, the choice of using an additive feature attribution method rather than a gradient-based method is mainly based on computational time. Indeed, during our tests, we estimated that the total computational time for processing MATRES using the integrated gradients (Sundararajan et al., 2017) method was three weeks compared to one day with KernelShap (Lundberg and Lee, 2017).

References

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26:832–843.

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

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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.

Chunkit Chan, Cheng Jiayang, Weiqi Wang, Yuxin Jiang, Tianqing Fang, Xin Liu, and Yangqiu Song. 2023. Chatgpt evaluation on sentence level relations: A focus on temporal, causal, and discourse relations. *ArXiv*, abs/2304.14827.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

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

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

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

James Gung and Jugal Kalita. 2012. Summarization of historical articles using temporal event clustering. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 631–635.

Vivek Gupta, Pranshu Kandoi, Mahek Vora, Shuo Zhang, Yujie He, Ridho Reinanda, and Vivek Srikumar. 2023. TempTabQA: Temporal question answering for semi-structured tables. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2431–2453, Singapore. Association for Computational Linguistics.

Kelvin Han, Thiago Castro Ferreira, and Claire Gardent. 2022. Generating questions from Wikidata triples. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 277–290, Marseille, France. European Language Resources Association.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

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

Nithish Kannen, Udit Sharma, Sumit Neelam, Dinesh Khandelwal, Shajith Ikbal, Hima Karanam, and L Subramaniam. 2023. Best of both worlds: Towards improving temporal knowledge base question answering via targeted fact extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4729–4744, Singapore. Association for Computational Linguistics.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Sha Li, Ruining Zhao, Manling Li, Heng Ji, Chris Callison-Burch, and Jiawei Han. 2023. Open-domain hierarchical event schema induction by incremental prompting and verification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5677–5697, Toronto, Canada. Association for Computational Linguistics.

Maofu Liu, Wenjie Li, and Huijun Hu. 2009. Extractive summarization based on event term temporal relation graph and critical chain. In *Information Retrieval Technology: 5th Asia Information Retrieval Symposium, AIRS 2009, Sapporo, Japan, October 21-23, 2009. Proceedings 5*, pages 87–99. Springer.

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. *arXiv preprint arXiv:1907.11692*.

Hector Llorens, Nathanael Chambers, Naushad UzZaman, Nasrin Mostafazadeh, James Allen, and James Pustejovsky. 2015. SemEval-2015 task 5: QA TempEval - evaluating temporal information understanding with question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 792–800, Denver, Colorado. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

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.

Yuanliang Meng, Anna Rumshisky, and Alexey Romanov. 2017. Temporal information extraction for question answering using syntactic dependencies in an lstm-based architecture. *arXiv preprint arXiv:1703.05851*.

Saumitra Mishra, Bob L Sturm, and Simon Dixon. 2017. Local interpretable model-agnostic explanations for music content analysis. In *ISMIR*, volume 53, pages 537–543.

Seyed Mahed Mousavi, Shohei Tanaka, Gabriel Roccabruna, Koichiro Yoshino, Satoshi Nakamura, and Giuseppe Riccardi. 2023. What's new? identifying the unfolding of new events in a narrative. In *Proceedings of the The 5th Workshop on Narrative Understanding*, pages 1–10, Toronto, Canada. 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, Austin, Texas. Association for Computational Linguistics.

OpenAI. 2023. Chatgpt.

James Pustejovsky, José Castano, Robert Ingria, Roser Saurı, Rob Gaizauskas, Andrea Setzer, Graham Katz, and D Radev. 2003a. Timeml: A specification language for temporal and event expressions. In *Proceedings of the International Workshop of Computational Semantics*, page 193.

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 *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

James Pustejovsky and Marc Verhagen. 2009. SemEval-2010 task 13: Evaluating events, time expressions, and temporal relations (TempEval-2). In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 112–116, Boulder, Colorado. Association for Computational Linguistics.

Dan Roth. 2000. Learning in natural language: Theory and algorithmic approaches. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Chao Shang, Guangtao Wang, Peng Qi, and Jing Huang. 2022. Improving time sensitivity for question answering over temporal knowledge graphs. In *Proceedings of the 60th Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 8017–8026.

Fei Song and Robin Cohen. 1988. The interpretation of temporal rdations in narrative. *AAAI-88 Proceedings, at Saint Paul*.

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.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Ronen Tamari, Chen Shani, Tom Hope, Miriam R L Petruck, Omri Abend, and Dafna Shahaf. 2020. Language (re)modelling: Towards embodied language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6268–6281, Online. Association for Computational Linguistics.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835, Toronto, Canada. Association for Computational Linguistics.

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

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, Atlanta, Georgia, USA. Association for Computational Linguistics.

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, Prague, Czech Republic. Association for Computational Linguistics.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

Xiaozhi Wang, Yulin Chen, Ning Ding, Hao Peng, Zimu Wang, Yankai Lin, Xu Han, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022. MAVEN-ERE: A unified large-scale dataset for event coreference, temporal, causal, and subevent relation extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 926–941, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yifan Wei, Yisong Su, Huanhuan Ma, Xiaoyan Yu, Fangyu Lei, Yuanzhe Zhang, Jun Zhao, and Kang Liu. 2023. MenatQA: A new dataset for testing the temporal comprehension and reasoning abilities of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1434–1447, Singapore. Association for Computational Linguistics.

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

Sen Yang, Xin Li, Lidong Bing, and Wai Lam. 2023. Once upon a *time* in *graph*: Relative-time pretraining for complex temporal reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11879–11895, Singapore. Association for Computational Linguistics.

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

Boyu Zhang, Hongyang Yang, Tianyu Zhou, Muhammad Ali Babar, and Xiao-Yang Liu. 2023. Enhancing financial sentiment analysis via retrieval augmented large language models. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pages 349–356.

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.

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

A Appendix

We report the tables regarding the label distribution of the three corpora (Table 6), the number of relations (Table 7), and the standard deviation for each in-context learning experiment done using five different few-shot prompts (Table 8). Furthermore, we provide the schema of the three types of prompts that we used in our experiments in Table 9. Table 11 presents the F1-scores for each class achieved using the best prompt settings. In addition to this, in Table 10 we provide a couple of wrongly predicted examples by the RoBERTa-based model which is challenging also for humans.

In this work, we have respected the original intended uses of datasets, models and any other artefacts.

A.1 Hyperparameters

For MATRES and TB-Dense we have chunked the text into sentences using NLTK[7] library as the corpus is natively split into paragraphs. The reason for this is to have minimal context for a given event.

Regarding the RoBERTa-based model, we experimented with different configurations and hyperparameters. In this, we have used one AdamW (Loshchilov and Hutter, 2017) optimizer for the encoder and another for the feed-forward layers (i.e. the classifier) with a learning rate of 1e-5 and 1e-4 respectively. Furthermore, we have used a linear scheduler on the optimizer of the encoder with warmup steps set to 10% of the total steps in training. In all the experiments the batch size is set to 8 event-pair data points.

In the in-context learning experiments (Brown et al., 2020), the number of few-shots, i.e. the number of ground truth examples used as context for the prediction, is set to one for each class of the corpus, i.e. five for TB-Dense and three examples for both MATRES and TIMELINE. The templates of the few-shot are replications of the prompt in the zero-shot version. An example of each different prompt can be found in Table 9. The examples have been extracted randomly from the training set of each corpus. To measure the impact on the performance of this selection, we have sampled and frozen five different sets to create the few-shots context for all three prompt types, i.e. $\boldsymbol{P}$, $\boldsymbol{QA_1}$ and $\boldsymbol{QA_2}$. In the few-shot context of $\boldsymbol{QA_1}$ and $\boldsymbol{QA_2}$ and at inference time $\boldsymbol{QA_2}$, we have kept the same order of the questions for all models and datasets which is *after*, *before*, *equal* and additionally for TB-Dense *includes* and *is included*. To fine-tune Llama2 7B, we have used a zero-shot approach as the model has to learn the task from the back-propagation of the error rather than the few-shots in the context. In this, we have used a learning rate of 1e-4 with AdamW (Loshchilov and Hutter, 2017) optimizer, training batch size 8, and we set the rank and alpha of LoRA to 32 and 64 respectively.

To fine-tune and test the RoBERTa model we used one NVIDIA GPU 3090Ti with 24GB. Regarding LLMs, we used four NVIDIA GPUs A100 with 80GB. The amount of GPU time needed to run all the experiments is around one month. To test the closed-sourced LLMs we have spent around $350 in API calls.

Temp. Rel.	MATRES	TIMELINE	TB-Dense
Before	58.0%	51.0%	42.0%
After	38.0%	47.0%	35.0%
Equal	4.0%	2.0%	3.0%
Includes	-	-	9.0%
Is Included	-	-	11.0%

Table 6: Label distribution of the three corpora computed on the entire three partitions (i.e. train, dev and test sets). *Simultaneous* in TB-Dense is mapped to *equal*.

Corpus	Train	Dev	Test
MATRES	9074	2133	724
TIMELINE	2384	284	685
TB-Dense	2008	375	789

Table 7: Number of relations over the three partitions for each corpus, without the class *vague*.

[7] https://www.nltk.org

Models	MATRES			TIMELINE			TB-Dense		
	P	***QA₁***	***QA₂***	***P***	***QA₁***	***QA₂***	***P***	***QA₁***	***QA₂***
Mistral 7B	13.9	7.4	4.7	15.3	4.0	8.5	1.0	0.1	0.3
Mixtral 8×7B	9.0	7.1	0.8	7.9	8.5	4.0	11.5	1.1	1.5
Llama2 7B	0.0	11.4	3.8	0.0	8.8	3.4	2.5	2.3	0.2
Llama2 13B	0.6	4.0	12.6	1.1	5.0	12.0	8.3	2.1	9.3
Llama2 70B	0.4	6.2	1.7	1.9	2.2	0.9	1.9	1.0	1.2
GPT-3	4.7	5.8	9.6	6.4	1.3	1.3	10.7	1.5	1.8
GPT-3.5	1.0	6.6	2.0	6.1	11.5	3.6	6.3	1.3	4.1

Table 8: Standard deviation computed on ten randomly generated prompts for the results achieved by LLMs with in-context learning (ICL) on MATRES (Ning et al., 2018), TB-Dense (Cassidy et al., 2014) and TIMELINE (Alsayyahi and Batista-Navarro, 2023) corpora. ***P*** refers to the example-label prompt. In ***QA₁*** and ***QA₂*** the TRC task is cast into two QA prompts. In ***QA₁*** the model answers one question at a time, while in ***QA₂*** the model uses as context its responses by answering the question in sequence.

Prompt types	Prompt
P	*Given the context:* It `[event1]` **accused** `[/event1]` the company of deliberately slashing oil revenues by overproducing oil and `[event2]` **driving** `[/event2]` down prices, among other charges. -> AFTER
QA₁	*Given the context:* It `[event1]` **accused** `[/event1]` the company of deliberately slashing oil revenues by overproducing oil and `[event2]` **driving** `[/event2]` down prices, among other charges. *Answer the question:* Does `[event1]` **accused** `[/event1]` happen after `[event2]` **driving** `[/event2]`? YES
QA₂	*Given the context:* It `[event1]` **accused** `[/event1]` the company of deliberately slashing oil revenues by overproducing oil and `[event2]` **driving** `[/event2]` down prices, among other charges. *Answer the questions:* Does `[event1]` **accused** `[/event1]` happen after `[event2]` **driving** `[/event2]`? YES Does `[event1]` **accused** `[/event1]` happen before `[event2]` **driving** `[/event2]`? NO Does `[event1]` **accused** `[/event1]` happen at the same time as `[event2]` **driving** `[/event2]`? NO

Table 9: Prompt schema used in ICL. In QA_1 we report one of the questions only, but the schema is the same for the others.

Context	Gold	Prediction
Evana Roth told CNN in August she ***e1:*thought** her husband devised the plan after he was fired from his job in July. Her attorney, Lenard Leeds, said she had been unaware of the ruse before she ***e2:*uncovered** the e-mail correspondence.	BEFORE	AFTER
The US embassy in Moscow has voiced concern and ***e1:*asked** the Russian government for an explanation. A new Russian law ***e2:*says** foreign-funded non-governmental groups (NGOs) linked to politics must register as "foreign agents" - a term which suggests spying.	AFTER	BEFORE

Table 10: Challenging examples mispredicted by LLMs and RoBERTa-based model. The examples are taken from the MATRES corpus.

Models	MATRES			TIMELINE			TB-Dense				
	Bef	Aft	Eq	Bef	Aft	Eq	Bef	Aft	Eq	Incl	Is_Incl
Mistral 7B	71.4	1.7	1.9	64.0	2.4	5.2	5.7	0.0	3.3	3.4	0.0
Mixtral 8×7B	73.9	0.6	1.8	70.9	0.1	3.7	25.7	1.4	1.5	0.0	7.7
Llama2 7B	71.9	6.4	2.1	70.1	25.7	0.0	0.2	30.1	0.3	9.5	0.0
Llama2 13B	2.0	53.5	0.7	5.0	57.7	2.5	10.6	41.9	3.3	11.0	5.1
Llama2 70B	76.8	33.6	0.0	74.8	24.1	0.0	0.3	31.7	3.6	0.0	0.0
GPT-3	67.2	26.9	0.0	68.4	32.9	0.0	0.0	3.1	2.6	5.0	1.1
GPT-3.5	72.4	36.8	0.0	70.3	30.0	9.5	36.1	0.7	0.0	0.0	0.0
Llama2 $7B_{\text{Fine-tuned}}$	87.2	77.9	0.0	81.9	81.2	0.0	63.7	34.6	0.0	0.0	0.0
Llama2 $13B_{\text{Fine-tuned}}$	88.6	82.1	0.0	68.3	51.8	0.0	65.8	51.9	0.0	0.0	0.0
RoBERTa	91.8	86.2	0.0	89.8	87.4	8.9	88.6	86.7	0.0	56.6	59.2

Table 11: F1-scores in percentage for each class of each dataset achieved using the best prompt settings. (Bef=Before, Aft=After, Eq=Equal, Incl=Includes, Is_Incl=Is Included)