TIMELINE: Exhaustive Annotation of Temporal Relations Supporting the Automatic Ordering of Events in News Articles

Sarah Alsayyahi and **Riza Batista-Navarro**
Department of Computer Science
University of Manchester
{sarah.alsayyahi,riza.batista}@manchester.ac.uk

Abstract

Temporal relation extraction models have thus far been hindered by a number of issues in existing temporal relation-annotated news datasets, including: (1) low inter-annotator agreement due to the lack of specificity of their annotation guidelines in terms of what counts as a temporal relation; (2) the exclusion of long-distance relations within a given document (those spanning across different paragraphs); and (3) the exclusion of events that are not centred on verbs. This paper aims to alleviate these issues by presenting a new annotation scheme that clearly defines the criteria based on which temporal relations should be annotated. Additionally, the scheme includes events even if they are not expressed as verbs (e.g., nominalised events). Furthermore, we propose a method for annotating all temporal relations—including long-distance ones—which automates the process, hence reducing time and manual effort on the part of annotators. The result is a new dataset, the TIMELINE corpus, in which improved inter-annotator agreement was obtained, in comparison with previously reported temporal relation datasets. We report the results of training and evaluating baseline temporal relation extraction models on the new corpus, and compare them with results obtained on the widely used MATRES corpus.

1 Introduction

Understanding the temporal structure of events in text is essential for a wide range of natural language processing tasks, e.g., question answering, information retrieval and inference (Campos et al., 2014; Ng et al., 2014; Ning et al., 2019b). Often, however, there is no explicit temporal information associated with most of the events in news articles. For instance, in the sentence *"He **pointed** to the possibilities of new business models, products and ways of working that could have a dynamic impact on living standards."*, there is no temporal expression associated with the event *"pointed"* that conveys when exactly it occurred. The extraction of temporal relations, i.e., determining whether an event occurred before, after or at the same time as another event, makes it possible to capture the temporal sequence of events, even in cases where the text does not explicitly mention any temporal information with respect to an event (Ning et al., 2018, 2019b).

Extracting temporal relations relies heavily on the annotation scheme adopted, which determines the granularity of the types of extracted temporal relations (Lim et al., 2019). In existing temporal information-annotated datasets (Pustejovsky et al., 2003; Bethard et al., 2007; Verhagen et al., 2007; UzZaman et al., 2013; Cassidy et al., 2014; Reimers et al., 2016; Mostafazadeh et al., 2016; O'Gorman et al., 2016; Ning et al., 2018; Naik et al., 2019), many types of temporal relations are ignored, ill-defined or focussed only on specific types of events. In most datasets, only relations between events in the same or adjacent sentences are tagged (Naik et al., 2019). Such limitation is the main reason for losing more precise temporal information for almost half of the events (Reimers et al., 2016). In addition, low agreement between human annotators is a common issue and needs to be improved by making the annotation task more clearly defined (Styler et al., 2014; Ning et al., 2018). Our work seeks to address these issues by making the following contributions:

1. A novel annotation scheme with an unambiguous definition of the types of events and temporal relations of interest. We also provide a method for automatically identifying and annotating every possible temporal relation in a given document.

2. A new dataset called TIMELINE[1] consisting

[1]Available at https://github.com/Alsayyahi/TIMELINE

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 16336–16348
December 6-10, 2023

	Dataset	# of documents	# of relations	Relation window	Type of events	IAA
1	TimeBank (Pustejovsky et al., 2003)	276	6418	0,1,..,n	Events defined by TimeML guidelines	55%
2	TimeEval3 (UzZaman et al., 2013)	276	6418	0,1,..,n	Events defined by TimeML guidelines	55%
3	TimeBank-DENSE (Cassidy et al., 2014)	36	12715	0,1	Events defined by TimeML guidelines	64%
4	RED (O'Gorman et al., 2016)	95	4969	0,1	Events defined by Thyme-TimeML guidelines	41%
5	MATRES (Ning et al., 2018)	36	1637	0,1	Events defined by TimeML guidelines	84%
6	TDDiscourse (Naik et al., 2019)	36	7x Larger than TB-D	0,1,..,n	Events defined by TimeML guidelines	69%

Table 1: Comparison among most relevant temporal relation datasets in the literature.

of 48 news articles, whereby a higher inter-annotator agreement was obtained in comparison with previously published temporal relation datasets.

3. An empirical analysis and an ablation study demonstrating the extent to which the TIMELINE dataset supports the development of models for ordering events in news articles.

2 Related work

TimeBank is the first temporal information-labelled dataset to provide different types of temporal annotations (i.e., events, time expressions, and temporal relations) in news articles (Pustejovsky et al., 2003). However, there are two main issues with TimeBank: (1) the annotators tagged only temporal relations (referred to as TLINKs) which are considered as important (Pustejovsky et al., 2003), leading to sparse annotations; and (2) the scheme did not specify when two events should be paired up in a relation; as a result, inter-annotator agreement (IAA) was only around 55%. Similar to TimeBank is the TimeEval3 dataset as it is a cleansed version of the former; it was created mainly for the TempEval shared tasks (UzZaman et al., 2013). Meanwhile, the RED corpus considered different relations between events (e.g., temporal, coreference, causal and sub-event relations). It is a rich dataset created mainly to support the development of multi-task systems; the IAA for the relations of interest (e.g., "before" relations) is relatively low, i.e., around 41% (O'Gorman et al., 2016).

TimeBank-DENSE is a subset of TimeBank, which was introduced to address the sparsity issue in TimeBank by annotating all possible event pairs, and all event and time expression pairs in each given sentence and its surrounding sentences (Cassidy et al., 2014). However, many ill-defined temporal relations were annotated, leading to low IAA. The MATRES corpus tried to solve this issue by adopting a scheme that takes into consideration multiple timelines, i.e., *axes*, distinguishing between events that actually happened (which belong to the main axis) and those which are only hypothetical (which belong to an axis parallel to the main one), for example. This multi-axis scheme required that each event relation is annotated while considering the relevant axis, thus improving the IAA significantly (84%) (Ning et al., 2018). However, they focussed only on events centred on verbs and ignored nominalised events.

Cheng and Miyao (2018) proposed automatically annotating temporal relations between events in a sentence and its surrounding sentences, using predefined rules based on the events' time anchors. They annotated temporal relations based on an existing dataset where the time anchors for events are already labelled (Reimers et al., 2016). Naik et al. (2019) suggested a heuristic algorithm for the automatic inference of relations using the corpus developed by Reimers et al. (2016). Moreover, they made the first attempt to capture long-distance relations by asking experts to manually annotate a subset of unlabelled long-distance relations based on textual cues, external knowledge and narrative ordering. However, state-of-the-art models perform worse on their dataset, TDDiscourse, compared with other datasets. Error analysis shows that the models failed to deal with some of the phenomena in their dataset (e.g., negated/conditional events, event coreference, and the requirement to have access to real-world knowledge).

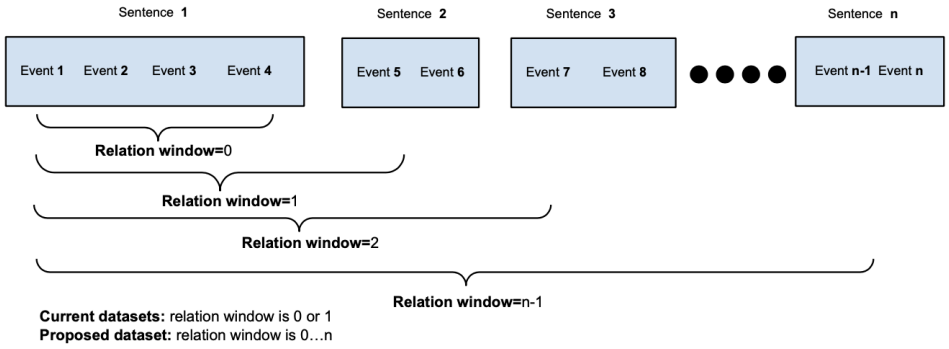

Figure 1: Illustration of the temporal relation window

Event type	Category
Intension, Opinion	On an orthogonal axis
Hypothesis, Generic	On a parallel axis
Negation	Not on any axis
Static, Recurrent	Other

Table 2: Different event types that are not in the main axis.

3 Motivation

This work is motivated by earlier relevant studies in the literature; we refer the reader to Table 1 for more information about previously proposed temporal relation datasets. We, however, attempted to address the shortcomings of the previously proposed annotation schemes and developed a new dataset that specifies the relative order of events mentioned in a given news article. In designing our annotation scheme, we considered the following questions:

1. What types of events will be included?
2. Is it possible to annotate the relations automatically based on the time anchors of events, and subsequently allow for retrieving the temporal order of any two events?

We reviewed existing temporal relation annotation schemes (Minard et al., 2015; Speranza and Minard, 2014) to answer the first question. We then decided to discard events that cannot be anchored onto a timeline; these include intended, negated events and events involved in conditional constructions. Such events are the source of many ill-defined relations (e.g., vague relations) in existing datasets. A specific temporal relation between two events is labelled *vague* if there is not enough information about the two events in the text that makes the annotator decide if the first event occurred before or after or at the same time as the second event. Consider the following example sentence: *"She **planned** to **attend** the conference yesterday."* The temporal relation between the two events (*"planned"* and *"attend"*) is vague as we cannot confidently determine the temporal relation between them based on the context alone, i.e., it is possible that the event centred on *"attend"* did not occur.

According to Ning et al., (2018), events belong to different time axes, hence the distinction between the following axes: (1) the main axis, i.e., a horizontal line where events that actually happened are represented (e.g., the event *"planned"* in the example sentence); (2) an orthogonal axis (a vertical line that is orthogonal to the main axis) where opinions/intentions are placed (e.g., the event *"attend"* in the same example); and (3) a parallel axis (a horizontal line parallel to the main axis) where generic and hypothetical events are placed. Hence, we focussed only on all events that belong to the main axis (events in the main storyline). We refer the reader to Table 2 for examples of events that do not belong to the main axis. We discuss details of how we identified events that need annotation in Section 4.2.

Regarding the second question, we concluded that annotating every possible temporal relation in a specific news article is a non-feasible task (Naik et al., 2019). Importantly, inconsistent temporal relation annotations are to be expected from a human annotator (e.g., a transitive constraint is not always satisfied) and have been noted in the TimeBank corpus (Bethard et al., 2007; Pustejovsky and Stubbs, 2011). Additionally, employing crowdsourcing for the annotation of these relations is expensive. For instance, Ning et al. (2018) reported

that it costs about 400 USD to annotate temporal relations between events in a given sentence and its surrounding sentences in only 36 news articles. Also, Reimers et al. (2018) highlighted that considering long-distance relations is required to retrieve correct temporal information for 40 % of events in news articles. Therefore, to address these issues, we decided to automatically generate temporal relations and to directly infer consistent relations within different windows, i.e., relations between events which are separated by $0...n$ sentences. Further details on how temporal relations are generated will be given later in Section 4.2. Please refer to Figure 1 for an illustration of the relation window.

4 Dataset construction

In this section, we describe the process for collecting the documents included in our corpus. This is followed by a discussion of the details of our proposed annotation scheme.

4.1 Document collection

The *LexisNexis* library is an online resource that offers access to court cases, commentaries, handbooks and news articles, amongst others[2]. The library was used to retrieve a total of 48 news articles published in a UK newspaper: The Times (London). Table 3 presents the queries that we used to retrieve the articles.

No	Category	# of articles	Publication Period
1	"Covid-19" and "Economy"	16	March 2020 - Feburary 2021
2	"Covid-19" and "Vaccine"	16	
3	"Covid-19" and "Quarantine"	16	

Table 3: Queries used in retrieving the news articles in the TIMELINE corpus.

4.2 Annotation Scheme

Our scheme consists of multiple layers of annotation which are described below.

Event annotation. Events in our corpus were annotated according to the TimeML guidelines (Saurı et al., 2006), which define an *event* as a situation that occurs. Events are centred on one or more trigger words and can be expressed in different ways. This includes verbs, e.g., *"said"*, or phrasal verbs, e.g., *"woke up"*, as well as nominal events, e.g., *"World Cup"* or *"demonstration"*. We included all events that can be anchored onto a timeline as long as they belong to the main axis. However, as discussed in Section 3, we excluded specific types of events: intended, negated, static, generic, and hypothetical events. In the Appendix, we provide a complete list of the broad types of events that we excluded, alongside some examples.

Time anchor annotation. Drawing inspiration from previous studies, we adopted the use of the concept of narrative container (NC) in order to increase the accuracy of temporal relation annotation (Pustejovsky and Stubbs, 2011). NC is the default interval surrounding the document creation time (DCT) of an article, and provides an estimate of when a given event with no explicit time anchor, happened. It is affected by different variables related to text style and genre; for example, the NC value for newspapers is 24 hours, while that for weekly and monthly publications is a week and a month, respectively. Since our corpus consists of newspaper articles published on a daily basis, we can set the value of the narrative container to 24 hours—this was made clear to our annotators. Furthermore, annotators were provided with the DCT for every news article. Annotators were advised to use external and background knowledge if it helps them in providing more accurate time anchors. Where an event occurred over an interval, annotators were asked to provide the time anchor based on the start of the interval.

Earlier work which attempted to automatically generate temporal relations based on time anchors of events (Cheng and Miyao, 2018; Naik et al., 2019) were hindered by their reliance on the Event-Time corpus (Reimers et al., 2016). In this corpus, some events were given under-specified dates (e.g., *"after 1990-XX-XX"*) which made it difficult to form temporal relation annotations involving such events. In contrast, in our annotation scheme, events are always given explicit or implicit dates. Specifically, annotators were asked to enter the time anchor of the form *YYYY-MM-DD* for each event, by choosing one of six possible options based on the type of temporal information associated with the event. For instance, if the temporal information associated with the event is mentioned explicitly in the text (e.g., *"June 14, 2022"*), the annotator specifies *"2022-06-14"*. If the temporal information associated with the event is mentioned in the text

[2] https://www.lexisnexis.com/uk/legal/

in a vague manner (e.g., *"August"*), the annotator specifies *"2022-08-XX"*. We refer the reader to the annotation guidelines in the Appendix for a list of all possible options with examples.

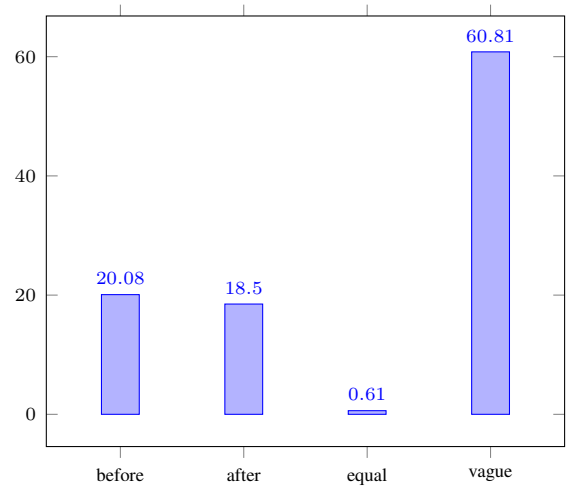

Figure 2: Label distribution in the TIMELINE corpus.

Temporal relation annotation. Before the automatic generation of relation annotations, the annotators were asked to answer a set of questions for each annotated event. These questions, for example, help determine whether two events happened at the same time (and thus should be given the "equal" label), and help reduce the number of "vague" relations by prompting the annotator to consider details within the context of events. We refer the reader to the annotation guidelines in the Appendix for a list of all the questions. Then, we developed a method for generating temporal relations that: (1) identifies every possible pair of events in a given document, and (2) generates consistent temporal relation labels based on the annotation given in the previous steps. The method handles every possible case to generate one of the following labels: `before`, `after`, `equal` and `vague` for each relation. For further details, we refer the reader to the Appendix, which shows the algorithm that generates a label for every possible relation. Figure 2 shows the distribution of the generated temporal relation labels. As illustrated in the figure, most of the relations are "vague" due to the inherent ambiguity of temporal information in natural language text.

5 Corpus Reliability

Three annotators contributed to the annotation of our corpus: the first one (the first author of this paper) annotated all the articles, whilst the second and third annotators annotated 31% of the articles. Table 4 presents the average inter-annotator agreement between the annotators at the level of events (calculated using F1-score) and temporal relations (calculated using micro-averaged F1-score and Cohen's Kappa). It is worth noting that the agreement over temporal relation annotations is based on events that annotators agreed on.

Inter-Annotator Agreement		
Event annotation	Relation annotation	
F1-score	Micro F1-score	Cohen's Kappa
91.29	92.98	86.75

Table 4: Inter-annotator agreement for event and temporal relation annotation.

	before	**after**	**equal**	**vague**	**TOTAL**
before	397	11	0	26	434
after	8	336	1	28	373
equal	2	0	10	2	14
vague	36	17	0	1268	1321
TOTAL	443	364	11	1324	2142

Table 5: Contingency matrix for temporal relation annotation. Rows correspond to Annotator 1 while columns correspond to Annotator 2.

The contingency matrix in Table 5 shows the agreement and disagreement between the first and second annotators with respect to temporal relation annotation. One can observe that the agreement between the annotators is high for all temporal relation types, which implies that the annotation scheme led to consistent annotations.

The second and third annotators are PhD Computer Science students who have received training on the proposed annotation scheme. Upon completion of the annotation tasks, they were compensated at an hourly rate of £15.

Three subsets were defined, containing randomly selected documents: training (70%), development (10%) and test (20%). We refer the reader to Table 6 for details on the number of documents and event pairs (annotated with temporal relations) in each subset.

6 Baseline methods

In order to assess the extent to which our proposed TIMELINE corpus supports the development of temporal relation extraction approaches, we sought to train and evaluate two baseline models for temporal relation extraction. Specifically,

Data splits	Number of documents	Number of pairs
Train	23	2384
Dev	11	284
Test	14	685

Table 6: Number of documents and event pairs annotated with temporal relation types in the TIMELINE corpus.

	Baseline method 1			Baseline method 2		
	P	**R**	**F1**	**P**	**R**	**F1**
before	77.59	86.89	81.97	79.03	86.14	82.43
after	78.57	86.51	82.35	79.03	86.14	82.43
equal	0	0	0	0	0	0
mi-F1	81.44			81.70		

Table 7: Performance of two baseline methods on the MATRES corpus (mi-F1 pertains to micro-averaged F1-score).

we employed two temporal relation classification models proposed by Han et al. (2019) as baseline methods. Both models are based on bidirectional long short-term memory networks (BiLSTMs), but one of them re-optimises the network to adjust for global properties, i.e., symmetry and transitivity constraints. These models were selected based on their highly competitive performance and the availability of their source code.

We note that prior to training and evaluating each of the said models, all temporal relations labelled as `vague` in the TIMELINE corpus, were discarded for the following reasons: (1) this type serves as a catch-all category for any relations which are ambiguously expressed in text and yet is over-represented (accounting for 60.81% of the annotated relations); and (2) more importantly, the performance for the `vague` relation type was not considered in previously reported work—they treated this label similarly to how they handled events with no temporal relations between them (Ning et al., 2019a).

In preparation for training the models, we generated BERT embeddings (Devlin et al., 2018) and part-of-speech (POS) embeddings for every token in the sentences containing events that are involved in a temporal relation, as both models take these as input representation. For the training process, we adopted the hyperparameter values used by Han et al. (2019).

	Baseline method 1			Baseline method 2		
	P	**R**	**F1**	**P**	**R**	**F1**
before	67.86	67.25	67.55	69.05	70.07	69.55
after	67.14	69.62	68.36	69.05	70.07	69.55
equal	100	5	9.52	0	0	0
mi-F1	67.51			69.05		

Table 8: Performance of two baseline methods on the TIMELINE corpus (mi-F1 pertains to micro-averaged F1-score).

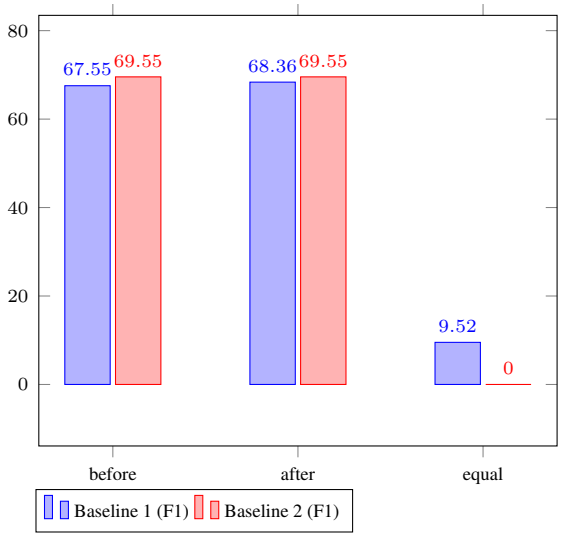

Figure 3: Performance of baseline models on the TIMELINE dataset in terms of F1-score.

7 Evaluation Results and Ablation Study

Table 7 and 8 show the performance of the baseline models on the MATRES and TIMELINE datasets, respectively. Han et al. (2019) reported slightly different performance obtained by both models on MATRES. They discussed in their paper that they used three random seeds; however, since the value of these seeds were not made available, we have been unable to replicate the same results.

As one can observe in Table 8, employing the second baseline model that adjusts for global constraints, leads to a performance improvement of 1.54 percentage points. This is slightly higher than the improvement (0.26 percentage points) obtained by the second model on the MATRES corpus. This is likely due to the higher number of globally consistent temporal relations in TIMELINE.

In the subsections below, we discuss the main differences between the two corpora, MATRES and TIMELINE, and the impact of each key difference on the performance of the baseline models.

	Baseline method 1			Baseline method 2		
	P	**R**	**F1**	**P**	**R**	**F1**
before	70.07	69.39	69.73	71.09	71.92	71.51
after	69.37	71.53	70.44	71.09	71.92	71.51
equal	100	8.33	15.38	0	0	0
mi-F1	69.74			71.09		

Table 9: Ablation study on Split 1A: the first split of the TIMELINE test set which contains relations involving verb-centred events.

	Baseline method 1			Baseline method 2		
	P	**R**	**F1**	**P**	**R**	**F1**
before	60.86	60.49	60.68	61.44	62.96	62.19
after	60.23	63.58	61.86	61.44	62.96	62.19
equal	0	0	0	0	0	0
mi-F1	60.54			61.44		

Table 10: Ablation study on Split 1B: the second split of the TIMELINE test set which contains relations involving non-verb-centred events.

	Baseline method 1			Baseline method 2		
	P	**R**	**F1**	**P**	**R**	**F1**
before	66.75	66.08	66.41	67.48	68.84	68.15
after	65.94	69.09	67.48	67.48	68.84	68.15
equal	100	6.25	11.76	0	0	0
mi-F1	66.37			67.48		

Table 11: Ablation study on Split 2A: the first split of the TIMELINE test set containing temporal relations between events separated by at most 4 sentences.

	Baseline method 1			Baseline method 2		
	P	**R**	**F1**	**P**	**R**	**F1**
before	69.45	68.95	69.20	70.25	70.75	70.50
after	68.90	70.39	69.64	70.25	70.75	70.50
equal	0	0	0	0	0	0
mi-F1	69.17			70.25		

Table 12: Ablation study on Split 2B: the second split of the TIMELINE test set containing temporal relations between events separated by more than 4 sentences.

7.1 Inclusion of non-verb events

News articles contain different events which, realistically, are not limited to events centred on verbs. Thus, we investigated the impact of including non-verb events on model performance, particularly on the F1-score for the `before`, `after`, and `equal` temporal relations. Specifically, we sought to assess whether a model has learned relations involving non-verb events to the same extent that it has learned relations involving verb-centred events.

To this end, we performed an ablation study by dividing the test set into two splits: (1) Split 1A contains samples with relations between verb events; and (2) Split 1B contains samples with relations where non-verb events are involved. We then evaluated the baseline models (trained and validated on the entire training/development sets) on each of the two splits. Unsurprisingly, we found that the performance on the first split (Table 9) is higher than the performance on the second split (Table 10). This indicates that the models are able to learn relations involving verb-centred events better than the relations with non-verb events during the training. This explains the higher performance of the models on the MATRES corpus given that it contains only verb-centred events. Moreover, one factor that may contribute to the reduced performance on the split that contains relations with non-verb events is that 70% of the relations in the training and development splits involve only verb-centred events.

7.2 Increasing the relation window

Retrieving the temporal relation between any two events (regardless of how far they are from each other in a given news article) is an essential requirement for different tasks and domains. The three following cases are a good illustration of the importance of considering such types of relations.

Case 1: In question answering (QA) tasks, to answer a specific time-based question, it is often necessary to retrieve the temporal relationship between an event in one of the first sentences and an event in the last few sentences of a given news article. It is impossible to retrieve this kind of long-distance relation in the previously published temporal information-annotated datasets since it is not tagged or cannot be retrieved using temporal reasoning (e.g., using transitive inference).

Case 2: In the medical domain, to extract useful information (e.g., a timeline of medical events) from clinical notes and reports, it is important to identify temporal relations between events that are not in subsequent sentences.

Case 3: Extracting a timeline of events from news articles allows decision-makers to conduct fine-grained analysis of these events; it is possible that events of interest are not in adjacent sentences.

We set out to investigate the impact of increasing

the relation window on model performance, particularly on the F1-score for the `before`, `after` and `equal` temporal relations. Specifically, we seek to determine if the models learned long-distance relations to the same extent as short-distance temporal relations.

To this end, we conducted an ablation study based on the test set subdivided into two splits: (1) Split 2A contains examples with short-distance relations, i.e., relation window $<= 4$, and (2) Split 2B contains examples with long-distance relations, i.e., relation window > 4. We set the threshold to 4 considering that the average relation window in our corpus is 9. If we split the set of relations in the corpus in this way, the short-distance relations involve events with 0 to 4 sentences between them; the long-distance ones involve events with more than 4 sentences between them.

The trained baseline models were then evaluated on each of the two splits. Interestingly, the performance on the second split (Table 12) is higher than on the first split (Table 11). This demonstrates that the models have learned long-distance relations better than short-distance temporal relations during the training process. A contributing factor to this is the fact that Split 2A has a slightly larger percentage of non-verb events, which we now know are more difficult for the models to learn (26.11% of the relations), compared with Split 2B (21.51% of the relations).

Datasets	Splits	# of possible pairs	# of non-vague pairs annotated	%
MATRES	Train	14219	943	6.63
	Dev	916	198	21.61
	Test	1433	272	18.98
	Total	16568	1413	**8.52**
TIMELINE	Train	5215	2384	45.71
	Dev	1313	284	21.63
	Test	2028	685	33.78
	Total	8556	3353	**39.19**

Table 13: Number and proportion of annotated temporal relations in MATRES and TIMELINE.

7.3 Extracting more relations

In an earlier section, we have shown that models find it more challenging to learn relations involving non-verb events, compared with verb-centred events. Despite the lower performance of the two baseline models on our proposed dataset, the models are able to extract more temporal relations than in MATRES.

Datasets	Splits	# of possible pairs	# of non-vague pairs classified correctly	%
MATRES	Train	14219	-	-
	Dev	916	-	-
	Test	1433	235	**16.39**
	Total	16568	-	-
TIMELINE	Train	5215	-	-
	Dev	1313	-	-
	Test	2028	473	**23.32**
	Total	8556	-	-

Table 14: Number and proportion of temporal relations in MATRES and TIMELINE that were automatically extracted by the better baseline model (the second model).

As shown in Table 13, in MATRES, only 8.52% of the possible relations were annotated as non-vague; meanwhile, 39.19% of the possible relations were labelled as non-vague in TIMELINE.

In Table 14, we show that the second baseline model extracted only 16.39% of the possible relations in the test set of the MATRES dataset. In our proposed dataset, TIMELINE, the model was able to extract 23.32% of the possible relations in the test set.

8 Reasoning Behind this Annotation in the LLMs Era

We believe that despite the advent of large language models (LLMs), this kind of fine-grained annotation is still necessary to support the development of supervised models. We argue that the temporal relation extraction performance of an LLM such as ChatGPT, for example, is not comparable in relation to that of supervised models. Firstly, Yuan et al. (2023) investigated ChatGPT's capability in zero-shot temporal relation extraction and showed that ChatGPT's performance is lower by up to 30% in terms of F1-score compared to supervised methods. Furthermore, we investigated the extent to which ChatGPT can extract temporal relations by prompting it using the zero-shot prompt proposed by Yuan et al. (2023) to identify temporal relations between events in the TIMELINE test set. Overall, ChatGPT obtained precision, recall and F1-scores of 31.11%, 35.67% and 33.24%, respectively. These are substantially lower than those of the second

baseline method, which obtained 69.05% for precision, 69.05% for recall and 69.05% for F1-score.

9 Potential Applications

Our annotation scheme and dataset hold promise for various practical uses. Extracting temporal relations from news articles can support information extraction applications such as automatic timeline extraction and question answering (QA). Moreover, considering that the focus of the dataset is on events on the main axis (i.e., events in the main storyline), this work can potentially support narrative extraction applications such as the analysis of events related to financial markets and event monitoring, e.g., in the context of disaster management (Norambuena et al., 2023).

10 Conclusion

In this paper, we present a new corpus, TIMELINE, which was annotated following a novel annotation scheme whereby non-verb-centred events are included, as well as long-distance temporal relations between events. The corpus was used in training and evaluating two baseline temporal relation extraction models. Based on our evaluation results, we assessed the impact of increasing the relation window and including non-verb-centred events on model performance. In addition, we demonstrated how our annotation scheme can support the development of models that can extract more relations in comparison with earlier datasets. In the future, we aim to increase the size of the dataset and employ it in a timeline generation task.

Limitations

This temporal relation research focussed on a specific type of publication, namely, newspapers articles published on a daily basis. As a result, we did not consider other types of publications which are published weekly or monthly. The primary motivation for this consideration is to use the narrative container concept (Pustejovsky and Stubbs, 2011), which has helped significantly to increase our annotation accuracy. Also, as we mentioned previously, we considered only events that can be anchored onto a timeline and that belong to the main axis (storyline).

References

Steven Bethard, James H Martin, and Sara Klingenstein. 2007. Timelines from text: Identification of syntactic temporal relations. In *International Conference on Semantic Computing (ICSC 2007)*, pages 11–18. IEEE.

Ricardo Campos, Gaël Dias, Alípio M Jorge, and Adam Jatowt. 2014. Survey of temporal information retrieval and related applications. *ACM Computing Surveys (CSUR)*, 47(2):1–41.

Taylor Cassidy, Bill McDowell, Nathanel Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. Technical report, Carnegie-Mellon Univ Pittsburgh PA.

Fei Cheng and Yusuke Miyao. 2018. Inducing temporal relations from time anchor annotation. 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 1833–1843.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Rujun Han, I Hsu, Mu Yang, Aram Galstyan, Ralph Weischedel, Nanyun Peng, et al. 2019. Deep structured neural network for event temporal relation extraction. *arXiv preprint arXiv:1909.10094*.

Chae-Gyun Lim, Young-Seob Jeong, and Ho-Jin Choi. 2019. Survey of temporal information extraction. *Journal of Information Processing Systems*, 15(4):931–956.

Anne-Lyse Myriam Minard, Manuela Speranza, Eneko Agirre, Itziar Aldabe, Marieke van Erp, Bernardo Magnini, German Rigau, and Ruben Urizar. 2015. Semeval-2015 task 4: Timeline: Cross-document event ordering. In *9th international workshop on semantic evaluation (SemEval 2015)*, pages 778–786.

Nasrin Mostafazadeh, Alyson Grealish, Nathanael Chambers, James Allen, and Lucy Vanderwende. 2016. Caters: Causal and temporal relation scheme for semantic annotation of event structures. In *Proceedings of the Fourth Workshop on Events*, pages 51–61.

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

Jun Ping Ng, Yan Chen, Min-Yen Kan, and Zhoujun Li. 2014. Exploiting timelines to enhance multi-document summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 923–933.

Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019a. An improved neural baseline for temporal relation extraction. *arXiv preprint arXiv:1909.00429*.

Qiang Ning, Hao Wu, and Dan Roth. 2018. A multi-axis annotation scheme for event temporal relations. *arXiv preprint arXiv:1804.07828*.

Qiang Ning, Ben Zhou, Zhili Feng, Haoruo Peng, and Dan Roth. 2019b. Cogcomptime: A tool for understanding time in natural language text. *arXiv preprint arXiv:1906.04940*.

Brian Felipe Keith Norambuena, Tanushree Mitra, and Chris North. 2023. A survey on event-based news narrative extraction. *ACM Computing Surveys*.

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

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

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

Nils Reimers, Nazanin Dehghani, and Iryna Gurevych. 2016. Temporal anchoring of events for the timebank corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2195–2204.

Nils Reimers, Nazanin Dehghani, and Iryna Gurevych. 2018. Event time extraction with a decision tree of neural classifiers. *Transactions of the Association for Computational Linguistics*, 6:77–89.

Roser Saurı, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. Timeml annotation guidelines version 1.2. 1.

Manuela Speranza and Anne-Lyse Minard. 2014. Newsreader guidelines for cross-document annotation nwr-2014-9.

William F Styler, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C De Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, et al. 2014. Temporal annotation in the clinical domain. *Transactions of the association for computational linguistics*, 2:143–154.

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

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

Chenhan Yuan, Qianqian Xie, and Sophia Ananiadou. 2023. Zero-shot temporal relation extraction with chatgpt. *arXiv preprint arXiv:2304.05454*.

Appendix

Annotation Guidelines

Step 1: Event annotation. All events according to the TimeML guidelines (Saurı et al., 2006) will be tagged, except for the following:

1. Cancelled or negated events will not be tagged; for example, *"He failed to **find** buyers"*, *"They don't **want** to play with us"*, or *"She cancelled the **meeting**"*. Moreover, uncertain events will not be annotated, e.g., *"We may **go**."*

2. Inspired by the TimeML guidelines, the following events will not be tagged: (1) generics (abstract and non-specific events), e.g., *"Fruit **contains** water."*, *"Lions **hunt** Zebra."*; (2) static events, e.g., *"New York **is** on the east coast."*

3. Hypothetical/conditioned events will not be annotated. For example, *"If I'm **elected** as president, I will **cut** income tax for everyone."*

4. Inspired by the annotation scheme followed by (Minard et al., 2015), adjectives express the property or attribute of an entity and anchoring them in time is not simple. Thus, adjectives will not be tagged.

5. Events after modal verbs will not be tagged. For example, *"We have to **leave**."*, or *"You must be sending the email by the end of the day."*

6. Intended events will not be tagged. They express intentions or things that are meant to happen or occur and signified by words such as *"plan"*, *"aim"*, *"intend"* and *"hope"*.

Step 2: Time anchor annotation. The annotators were asked to enter the time anchor for each event by choosing one out of six options:

- **Option 1**: If the text explicitly mentions the time of the event (e.g., *"Feb 1, 2021"*), the annotator should enter that date as a time anchor for the event. If the text does not mention the exact date but uses temporal expressions that are relative to the document creation time (DCT), e.g., *"today"*, *"last Friday"*, the annotator should use the calendar to enter the date in relation to the DCT.

- **Option 2**: If the text implicitly mentions the event's time (e.g., *"last August"*), the annotator should enter the date as a fuzzy date (e.g., *"2020-08-XX"*). Alternatively, if the text mentions that the event happened last year, the annotator should enter e.g., *"2020-XX-XX"*.

- **Option 3**: If the event has no temporal information, but it is clear from the text that the event happened around the document creation time (DCT), the date should be set to the default narrative container (NC) value for newspaper publications which is one day before the DCT.

- **Option 4**: If the event happens in the future, the default date will be one day after the DCT. Alternatively, if it is mentioned in the text that the event will happen sometime relative to the DCT, e.g., *"next Friday"*, the annotator can enter that day's date.

- **Option 5**: If the event happened in the past but the time is not mentioned in the text explicitly, the annotators can use any background or external knowledge to provide an accurate time anchor.

- **Option 6**: If the annotator understands from the text that the event did not happen around the document creation time, and the text does not provide any hints on when the event happened, the date should be entered as *"XXXX-XX-XX"*. Figure 4 shows how the events are represented in a timeline.

Step 3: Answer a set of questions for each annotated event.

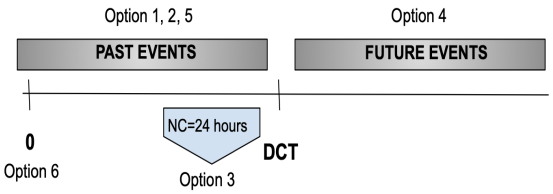

Figure 4: Timeline modelling for events

- **Question 1:** To annotate the relation between events that are the same (event coreference) with an `equal` label.

  Q1: *Does the event refer to another event in the document?* (Q1.a: Yes/No, Q1.b: event ID).

- **Question 2:** To annotate temporal relations with an `equal` label.

  Q2: *Did the event start or happen at the same time when another event in the same sentence happened?* (Q2.a: Yes/No, Q2.b: event ID).

- **Questions 3, 4 and 5:** To increase informativeness, i.e., to increase the number of non-vague relations.

  Q3: *Did the event happen on the same day as another event in the same sentence? If so, did the event happen at a different time compared with the other event?* (Q3.a: Yes/No, Q3.b: before/after, Q3.c: event ID)

  Q4: *Is this event with an unknown date (Option 6)? If so, did it happen before or after another event in the same sentence?* (Q4.a: Yes/No, Q4.b: before/after, Q4.c: event ID)

  Q5: *Were this event and another event in the same sentence given the same implicit time? If so, did this event happen before/after the other one?* (Q5.a: Yes/No, Q5.b: before/after, Q5.c: event ID)

- **Question 6:** To annotate the relation between events that happened around the DCT but were given different time anchors, as `vague`.

  Q6: *Did the event happen around the document creation time (e.g., within 24 hours)? (Yes/No)*

  For instance, consider the two events in the following sentences. Sentence 1: *"The pound gained almost 6 per cent against the dollar*

in July, ***approaching*** *$1.32 at one point yesterday before settling in evening trading at $1.31, up 0.04 per cent for the day and 5.7 per cent for the month.*". The event *"approaching"* happened *"yesterday"*. The temporal information is mentioned explicitly in the text for this event. Sentence 2: *"Bank of America Merrill Lynch strategists* ***said*** *the rest of 2020 could still see weakness for the pound as the period of August to December historically contains four negative months for sterling."* The event *"said"* happened possibly one day before the publication date (based on what the reader of the news article could infer according to the narrative container concept). However, it is not clear from the text which of the two events *"approaching"* or *"said"* happened first. Therefore, if the annotator answered the question with `Yes` for both events, our temporal relation generator will assign the relation label `vague` to these events to ensure accuracy.

- **Question 7:** To annotate a relation between events that are happening in the future but were given different time anchors, as a `vague` relation.

  Q7: *Is the event happening in the future? (Yes/No)*

  For instance, sometimes in the text, it is mentioned that some event (*Event 1*) will happen in the future without any time anchor; for another event (*Event 2*), the text says that it will occur at a specific time (e.g., *"next month"*). However, it might be unclear from the text which event will happen first. Therefore, if the annotator answered the question with `Yes` for both events, our temporal relation generator will assign the relation `vague` to these events.

Step 4: Temporal relation annotation. The temporal relations are annotated automatically based on Algorithm 1.

Special Cases

Below are two cases encountered during the annotation process that we needed to make the annotators aware of.

- **Event Coreference:** When we have more than two events referring to the same thing, the relations that involve these events have to be annotated manually after Step 4.
- **Subsequent events only:** In Q1, the annotators should verify that the event ID is associated with a subsequent event mentioned in the same or any following sentence. In Q2-Q5, the annotators should ensure that the event ID is associated with a subsequent event mentioned in the same sentence.

Algorithm 1: Temporal Relation Generation Method

M = all possible event pairs(ai, bi) in the document

for i in M **do**

if $((time(ai) == time(bi))$ **and** $(((Q1.a(ai) == \texttt{Yes})$ **and** $(Q1.b(ai) == bi))$ **or** $((Q2.a(ai) == \texttt{Yes})$ **and** $(Q2.b(ai) == bi))))$ **then**

$Label = \texttt{equal}$

else if $(((time(ai) < time(bi))$ **and** $(((Q6(ai) \neq \texttt{Yes})$ **or** $(Q6(bi) \neq \texttt{Yes}))$ **and** $((Q7(ai) \neq \texttt{Yes})$ **or** $(Q7(bi) \neq \texttt{Yes}))))$ **or** $((time(ai) == time(bi))$ **and** $(Q3.a(ai) == \texttt{Yes})$ **and** $(Q3.b(ai) == \texttt{before})$ **and** $(Q3.c(ai) == bi))$ **or** $((Q4.a(ai) == \texttt{Yes})$ **and** $(Q4.b(ai) == \texttt{before})$ **and** $(Q4.c(ai) == bi))$ **or** $((Q5.a(ai) == \texttt{Yes})$ **and** $(Q5.b(ai) == \texttt{before})$ **and** $(Q5.c(ai) == bi)))$ **then**

$Label = \texttt{before}$

else if $(((time(ai) > time(bi))$ **and** $(((Q6(ai) \neq \texttt{Yes})$ **or** $(Q6(bi) \neq \texttt{Yes}))$ **and** $((Q7(ai) \neq \texttt{Yes})$ **or** $(Q7(bi) \neq \texttt{Yes}))))$ **or** $((time(ai) == time(bi))$ **and** $(Q3.a(ai) == \texttt{Yes})$ **and** $(Q3.b(ai) == \texttt{after})$ **and** $(Q3.c(ai) == bi))$ **or** $((Q4.a(ai) == \texttt{Yes})$ **and** $(Q4.b(ai) == \texttt{after})$ **and** $(Q4.c(ai) == bi))$ **or** $((Q5.a(ai) == \texttt{Yes})$ **and** $(Q5.b(ai) == \texttt{after})$ **and** $(Q5.c(ai) == bi)))$ **then**

$Label = \texttt{after}$

else

$Label = \texttt{vague}$

end if

end for