Towards Consistent and Informative Timeline Generation

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computational Linguistics
by Research

by

Priyank Modi
20171055
priyank.modi@research.iiit.ac.in

International Institute of Information Technology
Hyderabad – 500 032, INDIA
July 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "*Towards Consistent and Informative Timeline Generation*" by *Priyank Modi*, has been carried out under my supervision and is not submitted elsewhere for a degree.

__________	____________________
Date	Adviser: *Prof. Manish Shrivastava*

To Mom and Dad...

Acknowledgments

I would like to express my deepest gratitude to everyone who has supported and guided me throughout my thesis journey.

I am immensely grateful to my advisor, Manish Shrivastava, for his passion, guidance, and encouragement. Our stimulating discussions have greatly contributed to my growth as a researcher.

I would also like to extend my thanks to IIIT-Hyderabad and its faculty members, especially Radhika Mamidi, for their mentorship and guidance throughout this journey.

My family and friends have been a pillar of support for me in this process, I can't aptly express my gratitude towards my parent, my brother Mayank, and my friends, Shantanu, Souvik, Ujwal (also my co-author), Aashna, Suryansh and Debojit. They have truly made this a memorable journey and helped me learn a lot along the way.

I would also like to acknowledge the efforts of my co-authors, Pranav, Alok, Maneesh Singh and Yatin Nandwani.

Lastly, I would like to extend my appreciation to the anonymous reviewers of our papers, whose constructive feedback and insightful suggestions have greatly enhanced the quality of our work.

Abstract

Understanding temporal relationship between events form an important part of document analysis and help in many downstream tasks, like Question Answering (QA) and Timeline Generation. Timeline generation serves as a cornerstone in NLP as it enables the reconstruction and visualization of the chronological sequence of events within textual data. Effective timeline generation serves as a foundational step toward coherent document understanding. However, existing temporal relation extraction models face notable limitations due to issues prevalent in annotated datasets. These include ambiguous annotation guidelines leading to low inter-annotator agreement, the omission of long-distance relations across document sections, and a narrow focus solely on verb-centered events. In response, this thesis introduces a pioneering approach aimed at creating a comprehensive discourse-level temporal event ordering dataset.

Central to our methodology is the transformation of relation classification between event pairs from a local context to the creation of discourse-level timelines. Our novel approach incorporates the concept of multiple timelines within a discourse, distinguishing between the actual timeline containing realized events and hypothetical timelines housing potential occurrences. To facilitate this innovation, we developed DELTA 2.0 (DiscoursE Level event Timeline Annotation) by re-annotating the TimeBank-Dense dataset. This effort resulted in a substantial increase in the inter-annotator agreement score and the number of meaningful relations captured, surpassing the scope of existing datasets focused solely on local temporal relations.

Furthermore, to streamline and enhance the efficiency of timeline annotation, we devised a user-friendly annotation tool. Employing this annotated dataset, a publicly available adapted-version of the state-of-the-art model, TIMERS, was trained, showcasing its effectiveness in predicting discourse-level event temporal relations. Additionally, through a comprehensive reproducibility study, we evaluated leading-edge models in the domain of event-event temporal relation classification. Leveraging advanced language model-based architectures such as BERT and RoBERTa, our analysis revealed promising outcomes. Moreover, integrating Graph Neural Networks into our methodologies enhanced the representation of temporal information, substantially improving the accuracy of generated timelines.

This work signifies a transformative shift towards discourse-level analysis in temporal relation extraction from news articles. By establishing DELTA 2.0 and integrating sophisticated neural network methodologies, this research lays a robust foundation for advancing the understanding of temporal relations in natural language processing. Such advancements have the potential to significantly impact document understanding and a wide array of practical applications in the field. Finally, we extend our work to indic languages via a TimeML compliant Hindi Timebank.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Understanding the temporal relationships between events is a fundamental aspect of human language comprehension. We naturally follow the flow of time as we read or listen, piecing together the sequence of actions and occurrences within a narrative. This ability to grasp temporal relations is crucial in various tasks, from summarizing news articles to building timelines of historical events. However, for machines, accurately identifying and classifying temporal relations in text remains a significant challenge.

The task of Temporal Relation Extraction (TRE) aims to automatically determine the temporal ordering between event mentions within a document. In the realm of NLP, events are acts or occurrences with certain temporal features. The relationships between these events offer important insights into the temporal dynamics, causation, and narrative structure of textual data. TRE is a multifaceted task that involves several key steps aimed at understanding the chronological relationships between events mentioned in text. The process typically begins with (1) preprocessing the text to extract linguistic features and syntactic structures, followed by identifying and normalizing temporal expressions such as dates, times, and durations. (2) Once temporal expressions are recognized, events are extracted from the text along with their associated attributes, including event types and participants. (3) With events identified, the task proceeds to identify and classify the temporal relationships between pairs of events, categorizing them into before, after, simultaneous, or more complex temporal relations like overlap or subordination.

This task has been the subject of numerous attempts, utilising a variety of strategies such as statistical models, deep learning techniques, and rule-based systems. Rule-based systems utilise human-crafted language patterns and heuristics to recognise and categorise temporal relationships, whereas statistical models, such Support Vector Machines (SVMs) (Cortes and Vapnik (1995)) and Conditional Random Fields (CRFs) (Lafferty et al. (2001)), learn to anticipate relationships based on pre-established properties. Recently, by learning representations directly from unprocessed text input, deep learning models such as Transformer-based (Vaswani et al. (2017)) architectures, Recurrent Neural Networks (RNNs) (Sherstinsky (2020)), and Convolutional Neural Networks (CNNs) (O'shea and Nash (2015)) have demonstrated potential.

Despite the progress made in event temporal relation extraction, several challenges persist.

1. **Inaccuracy in event identification:** The accuracy of the timeline hinges on the ability to correctly identify event mentions and accurately classify their temporal relationships. Errors in these initial steps can lead to inconsistencies and inaccuracies in the final timeline.

2. **Limited Datasets:** The effectiveness of machine learning models heavily relies on the quality and quantity of training data. Unfortunately, the availability of large-scale, high-quality datasets specifically designed for TRE tasks remains limited. The lack of diverse and well-annotated data hinders the ability of models to generalize to unseen scenarios.

3. **Focus on Local Context:** Many existing TRE models primarily focus on the immediate context surrounding event mentions. While local context plays a role, temporal relationships often depend on broader discourse structures and the overall flow of information within a document. Models that fail to consider this broader context may struggle to accurately classify complex temporal relations.

Addressing these challenges requires advancements in data annotation, model architectures, and techniques for domain adaptation and temporal reasoning. Efforts to overcome these obstacles will lead to the development of more robust and domain-agnostic event temporal relation extraction systems. Accurate identification of temporal relations is valuable in tasks such as question answering using which systems that can understand event sequences and provide more informative answers to questions that require temporal reasoning; event summarization, and event timeline generation. With the advent of large-language models (LLMs), and their excellent performance on various NLP and visiion tasks, it was interesting to look at how LLMs do on a complicated task like TRE that requires understanding long document context. As discussed in Section 4.3, we see LLMs fall well-short of baseline methods on this task. This is just another source of validation for the importance of the contributions of this thesis.

1.2 Thesis Contributions

The major contributions of the thesis are described below. Through these efforts, this thesis seeks to contribute to the development of more robust and generalized models for temporal relation extraction, ultimately leading to a deeper understanding of how machines can reason about the flow of time within natural language text.

1. **DELTA 2.0** - A dataset capturing global temporal relations in a document
2. **An annotation schema and an Annotation tool** that solves the problem of low IAA scores and high volume of VAGUE relations.
3. **Adapted-TIMERS** - An open source version of the state-of-the-art model for predicting temporal relations.
4. Baselines and Adapted-TIMERS on DELTA 2.0
5. **Hindi TimeBank - An ISO-TimeML Annotated Reference Corpus**, to encourage research in this area for Indic languages as well.

1.3 Thesis Organization

This thesis is organized into 5 chapters, briefly described below:

1. Chapter 2 explores the related work in the field. I explore prior work on event temporal relations and timeline generation, including work on predicting temporal relations using both neural and non-neural methods.
2. Chapter 3 showcases the work on event-event relations. Amongst the many event-event relations that exist, I focus on temporal relations. We showcase our efforts in building DELTA 2.0: a dataset for discourse-level timeline generation. We also built a strong baseline based on RoBERTa [Liu et al. (2019)] to validate our dataset.
3. Chapter 4 showcases work on state-of-the-art models in temporal relation prediction through a reproducibility study. We share an open-source version of the sota model, TIMERS, and apply it to DELTA 2.0.
4. Chapter 5 focuses on our contribution to indic languages through a Hindi TimeBank: A Time-ML Annotated dataset.
5. Chapter 6 concludes the thesis and. We explore further work, the challenges present and the shortcomings of the approaches explored in this thesis.

Chapter 2

Related Work

2.1 Event-event temporal relations

The early 2000s marked a resurgence of interest in temporal reasoning within natural language processing (NLP). The ACL Workshop on Temporal and Spatial Reasoning (2001) (ws-(2001)) and the LREC workshop on Annotation Standards for Temporal Information in Natural Language (2002) highlighted the importance of temporal aspects in tasks like information retrieval and extraction.

A key turning point came with the development of TimeML specifications by [Pustejovsky et al. (2005)] and the creation of the TimeBank Corpus by [Pustejovsky et al. (2005)]. These advancements paved the way for significant research in temporal relation extraction, particularly focusing on the task of temporal relation ordering between events.

TimeML defines these temporal expressions as follows.

> “A temporal expression in a text is a sequence of tokens (words, numbers and characters) that denote time, i.e. they express a point in time, or a duration or a frequency.”

The TimeBank Corpus: A Foundation for Event-Event Relations
TimeBank (tim (2006)) serves as a cornerstone dataset for event-event temporal relation extraction. It annotates temporal relations between events (or temporal expressions) using a fine-grained scheme consisting of 14 labels called ”T-Links.” These T-Links can occur between two events or between an event and a temporal expression.

TimeML defines temporal expressions as sequences of words, numbers, or characters encoding time. Examples include dates (”3rd January 2000”), durations (”sixty minutes”), and frequencies (”fortnightly”). Additionally, non-grounded expressions like ”today” or ”last year” are also considered temporal expressions.

The 14 T-Link labels in TimeBank capture various temporal relationships between events and expressions. Here’s a breakdown of some key relations:

1. Before/After: These represent the basic linear ordering of events. For instance, ”The meeting started (event A) before the presentation began (event B).”

2. Includes/Is Included: These capture containment relationships. ”The conference (event A) included a workshop (event B)” means the workshop happened within the timeframe of the conference.

3. Holds During/Held During: These describe an event persisting throughout the duration of another. ”The party (event A) lasted for three hours (temporal expression)” indicates the party happening during the entire three-hour period.

4. Simultaneous: This signifies events happening at roughly the same time. ”She was laughing and crying (events)” could be marked as simultaneous if the distinction between their exact order doesn’t significantly impact the understanding of the text.

5. Identity: This refers to two mentions representing the same event. ”John entered the room (event A). He then greeted everyone (event B)” might have an identity relation between ”entered” and ”greeted” if they describe a single action.

6. Other Relations: Additional T-Links like ”Immediately Before/After,” ”Begins/Begun By,” ”Ends/Ended By” capture more nuanced temporal relationships, including the start and end points of events relative to temporal expressions.

While the TimeBank corpus played a pivotal role in advancing event-event temporal relation extraction, it also presented some limitations. TimeBank annotations only captured a subset of the possible T-Links between events. Annotators were instructed to focus on relations critical for understanding the document, resulting in only 6418 relations being annotated across 183 documents. This sparse annotation leads to a significant number of ”false negatives” where actual temporal relationships are present in the text but not captured in the annotations. Moreover, the distinction between certain T-Link labels, such as ”before” vs. ”immediately before” or ”begins” vs. ”during,” can be unclear. This ambiguity can lead to confusion among annotators and result in poor inter-annotator agreement.

Dense Annotation for Richer Datasets:
To address the limitations of sparse annotation, researchers explored creating datasets with denser event annotations. Here are some notable examples:

1. Temporal Directed Acyclic Graphs (TDAGs): This approach views text as a linear sequence of temporal segments, where each segment maintains internal temporal coherence. While the order is preserved within each segment, the ordering might vary between different segments. However, TDAGs [Bramsen et al. (2006)] still do not capture all possible temporal relations within a document.

2. TempEval and Joint Event Timeline: These efforts focused on annotating relations only between specific syntactic event pairs within a sentence or corpus (e.g., TempEval-1 [Verhagen et al. (2007)] and TempEval-2 [Verhagen et al. (2010)]) or extending annotations on existing corpora like ACE 2005 (Joint Event Timeline) [Walker et al. (2006)]. While these approaches provided denser annotations than TimeBank, they limited the scope of relation extraction.

TB-Dense: A Dataset with Dense Local Graph Annotation:
Recognizing the limitations of previous approaches, TimeBank-Dense (TB-Dense) [Cassidy et al. (2014)] emerged as one of the first datasets to annotate all possible relations between entities within a specific context window. This approach acknowledges the cost-prohibitive nature of annotating every single event pair across an entire document. Instead, TB-Dense focuses on dense annotation within local contexts, considering a window of neighboring sentences. Within this window, all temporal relations between entities (events, temporal expressions, and document creation time) are densely annotated. TB-Dense recognizes three types of entities for temporal relation extraction:

1. Events: Occurrences or happenings within the text (e.g., ”meeting,” ”presentation”).

2. Temporal Expressions (TIMEX): Explicit mentions of time within the text (e.g., ”3rd January 2000,” ”last week”).

3. Document Creation Time (DCT): The timestamp associated with the document's creation.

TB-Dense utilizes the same T-Link scheme as the original TimeBank corpus, encompassing 14 relation labels to capture various temporal orderings between events and temporal expressions. These relations include:

1. Basic Ordering: ”Before” and ”After” represent the fundamental linear ordering of events.

2. Containment: ”Includes” and ”Is Included” describe when one event occurs entirely within another event's timeframe.

3. Event-TIMEX Relations: ”Holds During” and ”Held During” capture an event persisting throughout the duration of a TIMEX expression.

4. Simultaneity: The ”Simultaneous” relation indicates events happening at roughly the same time.

5. Identity: This relation applies when two mentions refer to the same event (e.g., synonyms or paraphrases).

6. Nuances of Start/End Points: Relations like ”Immediately Before/After,” ”Begins/Begun By,” and ”Ends/Ended By” capture more specific temporal details, including the start and end points of events relative to TIMEX expressions.

TB-Dense selects 36 documents at random from the TimeBank corpus and annotates them, resulting in a total of 12,715 relations. As anticipated, the vague relation constitutes a significant portion of these statistics, accounting for 5,910 out of the 12,715 relations. Among these relations, 6,088 involve event-event connections.

While TB-Dense provides denser annotations compared to TimeBank, it intentionally ignores relations between events that fall outside the defined context window. This is a trade-off to achieve denser annotation within a manageable scope. The annotation process focuses on explicit temporal cues within the context window. TB-Dense might miss relations that rely on implicit world knowledge or require reasoning beyond the immediate textual context. To address these issues, new datasets like TDDiscourse (TDD) [Naik et al. (2019)] and MATRES [Ning et al. (2018b)] were created.

TDDMan: Exploiting Discourse Cues for Document-Level Relations

TDDMan (Time Discourse Directed Acyclic Matcher) aims to capture long-distance temporal relations beyond the confines of local context windows. It leverages discourse cues within text to infer implicit temporal ordering between events. TDDMan employs a discourse parser to identify relationships between sentences and clauses within a document. This parsing helps to understand the overall flow of information and how different parts of the text connect. Similar to TB-Dense, TDDMan utilizes Temporal Directed Acyclic Graphs (TDAGs) to represent the temporal structure of the document. However, unlike TB-Dense, TDDMan can build TDAGs that span the entire document, capturing long-distance dependencies between events.

While TDDMan offers a broader scope than TB-Dense, it relies heavily on the accuracy of the discourse parser. Errors in parsing can lead to misinterpretations of the temporal flow within the document. Additionally, the process of building TDAGs for entire documents can be computationally expensive for large texts. Since TDDMan focuses on manual annotation, this suggests a dataset size smaller than TB-Dense, which contains annotations for 183 documents.

TDDAuto: Automating Dense Annotation for Scalability

TDDAuto (Temporal Discourse Directed Acyclic Graph Automation) tackles the challenge of manual annotation in TB-Dense. It focuses on automating the process of creating TDAGs, aiming for scalability in generating densely annotated datasets. TDDAuto leverages a combination of labeled and unlabeled data. The labeled data provides guidance to the model, while the unlabeled data allows the model to learn and improve its ability to identify temporal relationships and construct TDAGs automatically. By automating the annotation process, TDDAuto offers the potential to create large-scale densely annotated datasets without the burden of manual labeling. This can be particularly beneficial for training models on more extensive textual data. However, the accuracy of TDDAuto heavily depends on the quality and quantity of labeled data used for training. A limited amount of labeled data can lead to an underperforming model with errors in relation identification and TDAG construction. Additionally, ensuring the quality of automatically generated annotations remains a challenge.

Both TDDMan (Time Discourse Directed Acyclic Matcher) and TDDAuto (Temporal Discourse Directed Acyclic Graph Automation) inherit the T-Link relation scheme from TimeBank and TB-Dense. This scheme consists of 14 relation labels that capture various temporal orderings between events and temporal expressions as explained before while describing TB-Dense.

MATRES: Bridging the Gap Between Explicit and Implicit Relations

Recognizing the limitations of focusing solely on explicit cues in TB-Dense, MATRES (Making Annotations of Temporal Relations Easier and more Systematic) [Ning et al. (2018b)] aims to bridge the gap between explicit and implicit temporal relations.

MATRES provides a flexible annotation framework that allows annotators to capture both explicit temporal expressions and implicit relations that rely on world knowledge or broader context. The annotation process incorporates background knowledge through the use of external resources like knowledge bases or commonsense reasoning tools. This allows annotators to consider implicit temporal aspects that might not be explicitly stated within the text.

Unlike TimeBank and TB-Dense, MATRES acknowledges that events can be related along multiple temporal axes. Imagine a news article describing a historical event like a war. The war itself might have a start and end date (linear time axis), but there might be other relevant events happening concurrently (a separate temporal axis). The multi-axial approach in MATRES offers a richer representation of temporal relationships within text. It allows for capturing complex event structures that go beyond a single linear order. To simplify the annotation process, MATRES focuses on annotating the temporal relations between the start points of events. This reduces complexity compared to considering both start and end points for each event.

Annotators are asked to follow During annotation, annotators consider two questions:

- **Q1**: Can T_1 occur before T_2?
- **Q2**: Can T_2 occur before T_1?

Let A_1 and A_2 represent the answers to Q_1 and Q_2 respectively. Based on these answers, four possibilities arise:

1. If both A_1 and A_2 are affirmative, the relation is categorized as vague, indicating a contradiction.
2. If both A_1 and A_2 are negative, the relation is denoted as ”equal”, signifying that neither event can precede the other.
3. If A_1 is affirmative and A_2 is negative, the relation between E_1 and E_2 is established as ”before”.

4. If A_1 is negative and A_2 is affirmative, the relation between E_1 and E_2 is specified as ”after”.

In total, MATRES comprises approximately 1000 relations, with 800 relations on the main axis and the remaining on the orthogonal axes.

Defining and using subordinate axes effectively requires careful consideration of the specific domain and task. Additionally, ensuring consistency in annotation across different axes can be more complex compared to the single-axis approach used in TimeBank and TB-Dense.

2.2 Temporal Relation Extraction & Event timelines

Temporal Relation Extraction (TRE) is a crucial task in Natural Language Processing (NLP) that aims to automatically identify the temporal order between events described in text. Imagine reading a news article that mentions a company announcing a new product launch, followed by a sentence about the company hiring additional staff. Without understanding the temporal relationship between these events, it’s difficult to grasp the complete picture. TRE helps machines process such text and determine that the staff hiring likely happened ”after” the product launch announcement. This task becomes even more valuable when building rich timelines.

By automatically classifying the temporal order of events within a document, TRE systems can create timelines that visually represent the flow of information. These timelines offer a clear and concise way to summarize events and their chronological relationships, making it easier to understand the overall narrative within a text source.

Baseline: Current neural models for TRE often rely on pre-trained language models (LMs) like RoBERTa (Liu et al. (2019)). These models are powerful tools for understanding language, but they typically extract features based on the ”local neighborhood” surrounding event mentions. In simpler terms, they focus on the immediate context around each event.

The baseline model described by Zhao et al. (2020a) provides a good starting point for TRE tasks. It utilizes a simple architecture with an MLP classifier on top of a pre-trained LM. While seemingly uncomplicated, this approach has achieved performance comparable to more complex models. The baseline feeds the LM a concatenation of sentences containing the two events, allowing it to capture some context. The LM then generates embeddings for each event, and these embeddings are combined before being fed to the classifier, which ultimately predicts the temporal relationship between the events.

This baseline serves as a benchmark for more sophisticated approaches. Researchers have built upon this foundation by developing models that incorporate additional features, such as long-range dependencies within the document or explicit reasoning about causality between events.

Multi-task Learning for TRE: Beyond the baseline model, researchers have explored various strategies to improve TRE performance. One approach involves incorporating related auxiliary tasks within a multi-task learning framework. For instance, Ballesteros et al. (2020) leverage event-timex relation extraction as an auxiliary task, achieving gains through scheduled multi-task learning. They further enhance performance by using a self-training approach, where their model generates annotations for unlabeled data, which is then used to further refine the model.

Another line of work focuses on jointly modeling event extraction and TRE. Han et al. (2019) employ a joint bi-LSTM model with Structured Support Vector Machine (SSVM) loss (Joachims et al. (2009)). However, enforcing consistency through Integer Linear Programming (ILP) during training significantly slows down the process. As mentioned earlier, Zhao et al. (2020a) demonstrate that a simpler approach can achieve comparable performance.

Wang et al. (2020) take a different approach, training a joint model for both temporal and causal relation extraction. They incorporate inter-task constraints alongside the standard temporal constraints. Notably, they employ probabilistic soft logic to convert these constraints into soft constraints on class probabilities, allowing for more efficient training compared to earlier methods relying on MAP inference. Additionally, they leverage knowledge bases like ConceptNet (Speer et al. (2017)) and TempProb (Ning et al. (2018a)) to extract features for their classifier, potentially enriching the model's understanding of temporal relationships.

These advancements highlight the ongoing effort to push the boundaries of TRE performance. Researchers are exploring various techniques, including multi-task learning, joint modeling, and incorporating external knowledge sources, to develop increasingly accurate and robust models for extracting temporal relations from text.

Non-neural methods: The previous section focused primarily on recent advancements in neural network-based approaches for Temporal Relation Extraction (TRE). However, it's important to acknowledge the significant contributions of non-neural and rule-based methods that have laid the groundwork for this field. Let's delve into these alternative approaches.

1. **Rule-Based Systems** One prominent approach in TRE utilizes rule-based systems. These systems function by applying hand-crafted rules that define patterns for identifying events and their temporal relationships within text. For example, a rule might be designed to recognize the phrase ”after the meeting” as indicating a subsequent event.

   The strength of rule-based systems lies in their ability to achieve high accuracy for well-defined patterns and specific domains. Additionally, they offer the advantage of interpretability. The reasoning process behind the rules is clear and understandable, allowing users to comprehend how the system arrives at its conclusions.

2. **Statistical Machine Learning Models** Another approach to TRE leverages statistical machine learning models. These models train on annotated TRE data to identify features

within text that are indicative of temporal relations. Imagine a model that learns to consider features like the presence of specific temporal keywords (”before”, ”during”) or the relative position of event mentions within a sentence.

Statistical machine learning models offer greater flexibility compared to rule-based systems. They can learn patterns from data and adapt to unseen examples that might not be explicitly covered by predefined rules. Additionally, these models have the potential to scale effectively when dealing with larger datasets.

However, these systems come with limitations. Developing and maintaining a comprehensive set of rules can be a laborious task. Researchers need to invest significant manual effort to create and refine rules that can handle various temporal expressions, event types, and syntactic structures within language. Furthermore, rule-based systems struggle to adapt to complex sentences, unseen variations in language use, and new types of temporal expressions that may emerge over time.

Statistical models aren’t easy either. Creating effective statistical machine learning models for TRE requires careful feature engineering. Researchers need to meticulously select and design features that capture the most relevant information about temporal relationships within the text. Furthermore, the performance of these models heavily relies on the quality and quantity of annotated training data. Limited or low-quality data can hinder the model’s ability to learn the nuances of temporal expressions and reasoning required for accurate TRE.

Recently, there has been little work using non-neural methods for this task looking at the results of approaches utilizing language models (LMs). Such approaches are still very insightful and provide usefuel cues while creating datasets.

Document level extractions: While local context approaches can handle cases with explicit temporal phrases (e.g., ”The meeting happened yesterday, followed by a press conference”), they struggle with document-level relationships. The limitations of local context approaches stem from their inability to capture long-range dependencies and implicit temporal cues within documents. These approaches treat each event mention in isolation, failing to consider the broader structure and relationships between sentences that can reveal the overall flow of events. In the quest for state-of-the-art performance in Temporal Relation Extraction (TRE), Mathur et al. (2021) introduced TIMERS, a model that goes beyond local context and leverages auxiliary information from multiple sources. While traditional approaches like Zhao et al. (2020a) rely solely on local embeddings generated by a language model (LM), TIMERS incorporates additional information to create a more comprehensive understanding of temporal relationships within text.

One key aspect of TIMERS is its ability to extract global temporal cues. It achieves this by employing a Graph Neural Network (GNN) that analyzes the entire document. This allows the model to identify relationships between events spread across sentences, providing a broader perspective on the temporal flow of information. Additionally, TIMERS utilizes a pre-trained model from Lee et al. (2017) to extract discourse-level features. These features capture the overall structure and relationships between sentences, potentially revealing implicit temporal order that might be missed by focusing solely on local context.

To further enrich its understanding of temporal relationships, TIMERS constructs a Temporal Graph (TG). This graph relies on two crucial pieces of information: annotated and canonicalized time expressions (TIMEX) and document creation time (DCT). Nodes within the TG represent events, TIMEX expressions, and the DCT. Edges are then added between these nodes based on their inferred temporal relationships. For instance, an edge would connect the DCT node to a TIMEX node representing a future event.

TIMERS further enriches the node embeddings within the TG by incorporating information about temporal arguments and hierarchical syntactic structure. The model utilizes a pre-trained Semantic Role Labeling (SRL) (Marcheggiani et al. (2017)) model to identify the temporal arguments associated with each event. These arguments might specify the time frame or duration of the event, and edges are added within the TG to connect event nodes to their corresponding temporal arguments. Additionally, a document-level hierarchical syntactic graph is constructed, capturing the relationships between words, sentences, and the overall document structure. The node embeddings within the TG are initialized with the output of a GNN operating on this hierarchical syntactic graph.

By combining information from local embeddings, global temporal cues, discourse features, and a rich temporal graph, TIMERS achieves a comprehensive understanding of temporal relationships within text. This comprehensive approach is what propels TIMERS to achieve state-of-the-art performance in TRE tasks.

2.3 TimeML adaptations

In response to the challenge of inconsistent temporal data representation across systems, ISO-TimeML (Pustejovsky et al. (2010)) emerged as an international standard. This standard built upon the groundwork laid by TimeML, a popular framework for annotating events, times, and their relationships within text. TimeML itself defines a comprehensive annotation scheme, tagging temporal expressions and eventualities (events, states, etc.) with various attributes. For events, the EVENT tag captures details like class, tense, and modality, while the TIMEX tag handles temporal expressions by resolving their values. TimeML further enables the specification of temporal relationships, subordinating relationships, and aspectual relationships to connect events and temporal expressions. These strengths made TimeML a natural foundation for ISO-TimeML. This international standard prioritizes language-agnostic application, defining an event simply as something relatable to another event or temporal expression using ISO-TimeML relationships. TimeBanks, built upon the TimeML annotation scheme, serve as these valuable resources. Let's delve into some specific examples:

French TimeBank (FTB): The French TimeBank (FTB) is a prominent resource for TRE tasks in French text. It was created through a collaborative effort (Bittar et al. (2011)) and leverages the TimeML annotation scheme to annotate events, temporal expressions (TIMEX), and their relationships within French documents. The FTB covers a variety of genres, including news articles, biographies, and historical texts, providing a diverse dataset for training TRE models. Researchers interested in French TRE tasks can find the FTB corpus and its annotation guidelines publicly available online

Italian TimeBank (It-TimeML): Similar to the FTB, the Italian TimeBank (It-TimeML) (Caselli et al. (2011)) is a resource specifically designed for TRE tasks in Italian text. It adheres to the ISO-TimeML annotation standard, ensuring consistency with other TimeBank efforts. Researchers involved in the creation of It-TimeML aimed to provide a high-quality, genre-balanced corpus for Italian TRE. The It-TimeBank encompasses various genres, including news articles, fiction, and weblogs, offering a well-rounded dataset for training models that can handle diverse text styles in Italian. The It-TimeBank corpus, along with its annotation guidelines, can be accessed online

Persian TimeBank (PerTimeBank): While TimeBanks exist for many European languages, resources for languages like Persian are still under development. The Persian TimeBank (PerTimeBank) is an ongoing project aiming to create a comprehensive corpus for TRE in Persian text. This initiative Yaghoobzadeh et al. (2012) utilizes the TimeML annotation scheme to annotate events, temporal expressions, and their relationships within Persian documents. The PerTimeBank is expected to contribute significantly to advancements in Persian TRE research. While the PerTimeBank is not yet publicly available in its entirety, ongoing research efforts hold promise for the development of this valuable resource.

Chapter 3

Temporal Relations

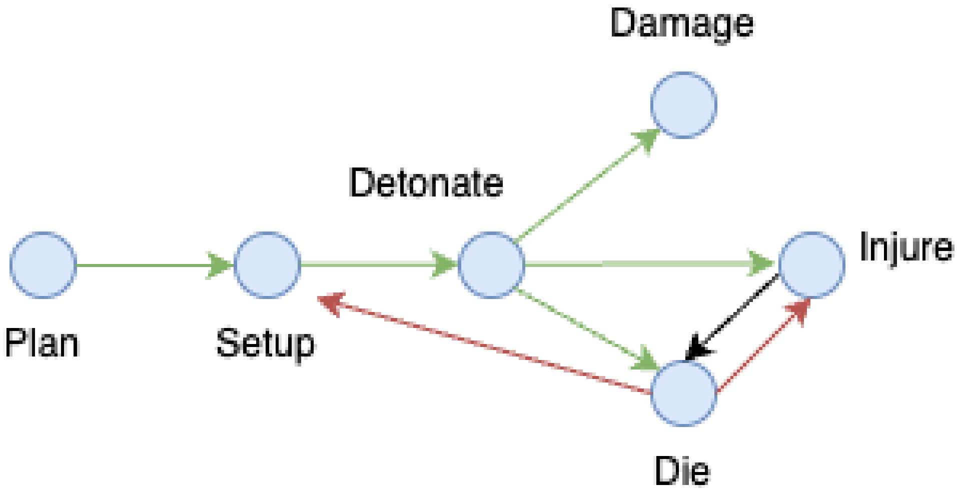
3.1 Introduction

The domain of Natural Language Processing (NLP) has witnessed a significant surge in the exploration of event identification and event relations. These identified relationships serve multifaceted purposes across diverse fields, ranging from facilitating tasks like question answering and summarization. This trend holds particular significance in the realm of crafting event timelines, wherein the temporal connections between various occurrences play a pivotal role. Nonetheless, one noteworthy finding in the array of datasets accessible for the purpose of extracting event linkages and creating timelines is the overwhelming emphasis placed on short-range temporal relationships. These linkages usually cover situations in which actions take place between adjacent sentences or within the same sentence (intra-sentential). These kinds of datasets have certainly advanced our understanding, but they also naturally restrict the useful applications of these kinds of systems. The reason for this constraint is that a significant fraction of temporal event relations really appear between sentences that are scattered throughout the speech; these are known as long-distance event temporal relations.

The accurate identification of temporal event linkages has been the focus of several efforts. But extending these linkages to discourse-level contexts is beyond the capabilities of the existing annotation schemas. We can summarize the major issues of existing datasets into the following 3 buckets:

1. Context Unawareness: Most existing datasets like MATRES Ning et al. (2018b), TDD Naik et al. (2019), Timebank only annotate relations between events in adjacent sentences or within the same sentence. The major challenge here is remembering long context while annotating. Think about this: an event that is introduced at the beginning of a document may simultaneously occur with an event that is presented towards the end of the document. This temporal disjunction highlights how complex creating a timeline at the document level is. However hard it might be, this leads to lossy timelines with little practical prowess.

2. Global Inconsistency: When creating temporal event graphs, local pairwise classification is likely to introduce competing predictions. An illustration of contradictory local predictions can be seen in Figure. Here the Red edges show incorrectly marked relations.



3. Ambiguous events and relations: The task of creating event timeline datasets has 2 important parts, identifying events and annotating the relations between events. Most datasets do not provide clear rules about either. For example, whether or not nominal events like 'imagine', " will be included; or; how should negative events be handled?

The major goal of our study was to identify the temporal correlations between occurrences at the discourse level, which goes beyond the boundaries of neighbouring phrases. By framing this work as an event timeline generation issue with a nuanced focus on capturing the broader narrative context, we propose a paradigm shift. Our goal is to improve the effectiveness and practicality of event timeline generation systems by exploring the complexities of long-distance event temporal linkages.

Within our model, we define a timeline as an organised chronological sequence of occurrences with respect to one another. Creating a timeline for a whole book is inherently complex, mainly because long-distance event linkages are so common. Our schema takes a page out of the ideas contributed by MATRES [Ning et al. (2018b)], which splits the document timelines into multiple axes. But MATRES only look at events between adjacent sentences, that too, on the primary axis. This leaves out a ton of relations. We solve this by introducing the idea of multiple timeline co-existing at the same time, where a primary or 'real' timeline, just like MATRES, includes all the events that have actually happened, and multiple other hypothetical timelines of events which includes events whose occurrence is not certain, exist in parallel. Of course, working with multiple different timelines is hard, even for expert annotators. In addition to that, our mission of annotating all event relations in the document makes it even harder to remember contextual information across distant events. Thus, in order to make annotation efforts efficient we also came up with an open-sourced multi-step annotation tool to enable annotators to not only ease the process of marking and visualizing event timelines for annotators, but also ensure global consistency of relations using a set of rules, so that annotators are instantly informed if they misunderstand certain events and make conflicting timelines.

We annotate the same set of documents as previous research, including TDD [Naik et al. (2019)] and MATRES [Ning et al. (2018b)], to provide a fair comparison. These files consist of a collection of English news stories. We generate a dataset containing around 45,000 temporal associations by annotating every pair of events in the documents.

To summarize, our major contributions are as follows:

- We extend the task of identifying temporal relations between event pairs to the discourse level and formulate the task as an event timeline generation task.
- We introduce a new annotation schema for event timeline generation. Our schema differs from existing schemas in a few ways:
  - We employ a multi-step annotation procedure that first defines an axis and then defines the relative ordering of events on that axis.
  - As we branch event timelines, we pay special attention to the real timeline which consists of events that have actually occurred.
  - At every step, we check for consistency of relations and auto-infer relations based on some conditions.
- We build and release a novel annotation tool which allows annotators to easily mark long distance event-event relations by offering a visual representation of the timeline and incrementally increasing the context length i.e. the distance between events to mark relations between.

3.2 Relations

We have a total of six relations, five of which were already part of DELTA 1.0, with an additional relation 'Equal' being added in place of indeterminate. The reason for deleting Intdeterminate was that it created a bias in the models due to it's volume. The only reason for adding Indeterminate in the first place was based on the idea of annotating every possible event pair relation. However, this has no real advantage, and simply makes the graph visually complicated. Our annotation tool internally captures whenever a chain of relations breaks due E_A and E_B having No Relation between them and infers other No Relations. All the six relations are defined with examples below:

3.2.1 Before

In temporal relations, if event A happens clearly before event B, we denote this as the `before` relation.

For instance, consider the example, ”Emma **baked** a birthday cake and **decorated** it with cherries.”. In this case, the event of baking (*baked*) precedes the event of decorating (*decorated*). Hence, the relation between these two events is labeled as `before`. Many cause-and-effect sequences exhibit this `before` temporal relationship.

3.2.2 Simultaneous

We adopt the TimeBank’s definition of Simultaneous relations [Pustejovsky et al. (2003)]. According to this definition, two events are labeled as `simultaneous` if they occur at the same time or if their timing is so close that distinguishing between them does not alter the temporal interpretation of the text.

For instance, in the sentence ”Emma ate a burger and drank a cup of coffee,” the events *ate* and *drank* are considered simultaneous. Although it’s possible that Emma either ate the burger first or drank the coffee first, this order doesn’t change the temporal understanding of the text, so we label these events as `simultaneous`.

3.2.3 During

An event pair is labeled with the *During* temporal relation when one event, denoted as A, takes place entirely within the duration of another event, denoted as B.

For instance, in the sentence ”Emma flew to London and drank a cup of coffee on the plane,” the event of drinking (*drank*) happens during the event of flying (*flew*). Therefore, the event *drank* is categorized as occurring DURING the event *flew*.

3.2.4 HET

The task of annotating event relations at a document level is hard. One of the reasons for this is that events within a document can vary in their certainty levels. Hence, when constructing an event timeline, it’s crucial to account for the certainty or modality associated with each event [Mitamura et al. (2015)]. On a broad level, events can be categorized as either ”real,” denoting those that have genuinely occurred, or ”hypothetical,” signifying events with uncertain occurrence status [Feldman et al. (1986); Ekdahl and Grimes (1964)]. This determination of certainty is known as event factuality prediction [Saurí and Pustejovsky (2009); Lee et al. (2015)].

In our discussion, we differentiate between two types of timelines: real and hypothetical. A ”real timeline” is a strict sequence that only includes events that have actually happened. Since the real world follows a single course of history, all true events inevitably fit into this singular real timeline. In contrast, "hypothetical timelines" allow for more freedom. Unconstrained by the limitations of the actual world, we can create multiple hypothetical timelines where different events, even those that never occurred, can be played out. This flexibility allows us to explore possibilities and imagine alternative scenarios.

When discussing hypothetical events, we often encounter verbs that use irrealis moods, signaling that the actions they describe are not factual but rather unreal [ELLIOTT (2000)]. These irrealis moods can take several forms. One common type is the subjunctive mood, which conveys wishes, desires, or imagined scenarios. Verbs like "dreamed" or "imagined" exemplify this usage. Another form is the dubitative mood, which expresses uncertainty or doubt. An example of this can be seen in the sentence "We believe Rooney scored the winner," where the speaker isn't entirely certain about the outcome. Additionally, future tense verbs and reported speech can also contribute to the construction of hypothetical events.

These irrealis verbs, termed anchors, link hypothetical timelines to real ones. The relation between an anchor and the closest verb (in terms of lexical distance) in the hypothetical timeline is known as HET. For instance, in the sentence "Dean thought that kicking the ball hard won't matter as the German goalkeeper would save it nonetheles.", the event "thought" serves as an anchor to a hypothetical timeline, for the relation between the events "kicking" and "save" to lie on the same lexically relevant timeline.

3.2.5 Vague

We use the VAGUE relation when both the events belong to the same timeline, but there is not enough information given in the document to ascertain the sequence of events. Consider the following example, "I ate cake for Harry's birthday last week. I also sold my bike after I crashed it last week." Both the events ate and sold happened last week, but since the ordering of events between ate and sold is not known, the temporal relation for this event pair will be marked as VAGUE.'. We also use a VAGUE relation if annotators degree on the event relation between a pair of events.

3.2.6 Equal

An event pair has the temporal relation *Equal* when both the events refer to the same occurrence i.e. the two events are co-referring to one another.

Consider the example, "Jack was **listening** to the Smiths on his way to the pool. However, enough sound was leaking from his earphones. Martha who was on the back seat could easily figure out what Jack was **listening** to." Here the 2 events listening refer to the same event and will be marked as equal. In DELTA 1.0, such events were being marked as simultaneous which led to confusion for annotators because this would sometimes span two different timelines and lead to inconsistencies.

DELTA 1.0 and it's Limitations

In DELTA 1.0, we chose a dense annotation schema i.e for every event pair in the document, there must be a temporal relation. The idea behind this was to ensure that no relation is missed out, which is quite likely when working in the long context of the document. In order to prevent annotators from annotating n^2 relations that can be really tiring, we chose to automatically infer new relations from existing relations based on a set of simple logical rules. These inferred relations were then added to the timeline graph. We also introduced the concept of multiple timelines, where annotators could mark events as they occur with multiple timelines occurring simultaneously. Then at the end of the document, they could review the different timelines, and add the relations to connect across timelines be it an additional relation such as before, during, simultaneous, HET or classifying the relation between the two timelines as indeterminate or vague.

While we saw some promising results with DELTA 1.0, there were a few limitations:

- **Low IAA scores:** Even with the annotation tool easing the annotation efforts and the automatic inference automatically populating new relations, it is still time-consuming and expensive to generate complete discourse level temporal relations between events. This is also reflected by the IAA score being on the lower end.

- **VAGUE relations:** In the attempt to ensure no event relations are missed out, DELTA 1.0 led to a lot of VAGUE and Indeterminate relations being generated. These relations are not helpful in generating timelines and also misleading while training models on such datasets as these relations lead to a lot of false-positive predictions due to their sheer volume.

- **No Event Co-reference:** DELTA 1.0 had no mechanism of resolve event co-reference. Whenever 2 events occurred in a similar timeframe, they were marked as simultaneous.

- **Locally-focused baseline:** We only built a context based encoder to classify the temporal relations. Since the context window is small compared to the total size of the document, it cannot get all the context required for the prediction of long distance relations.

3.2.7 Reasons for Low IAA in DELTA 1.0

3.2.7.1 Anchoring events to the correct timeline

DELTA 1.0 had a tiring, one-pass annotation strategy which made it hard to anchor events onto the correct timeline.

- The annotation methodology in DELTA 1.0 expected annotators to anchor an event to a timeline, and identify the temporal relations with other events(on the same or on different timelines) at the same time.
- This sometimes led to "real-world" events being anchored onto a hypothetical timeline or vice-versa.
- The cumbersome process also caused annotators to lose track of the context of the current sequence of events. Annotators found it easier to first identify events on a particular timeline(selecting from the set of non-anchored events), and then determining the temporal ordering of events on that timeline.
- There were no rules to anchor events on the "real" axis. In DELTA 2.0, this is solved by assigning time anchors(absolute time or relative to Document Creation Time) to events as a precursor of finding relations between events on that timeline. Note that all events on a hypothetical timeline will have a time anchor "HHHHHH-HH", making it easy to identify events part of the "real" timeline.

Consider this example from `Document #AP900816-0139`:

After a two-hour meeting at his Kennebunkport home with King Hussein of Jordan , Bush ***said*** *,"I did not* **come** *away with any feeling of hope that Iraq would withdraw its army from Kuwait."*

Both annotators incorrectly assigned the event *"come"* onto a hypothetical timeline in DELTA 1.0. This likely happened due to a trend in the other documents, where events following the reporting verb were either of "subjunctive" or "dubiative" moods. However, in DELTA 2.0, since the first step was to assign a time anchor to anchorable events without thinking about the temporal relations between two events, all 3 annotators assigned a time anchor - `17/03/2003` to the event *"come"*, thus, correctly marking it on the real axis.

3.2.7.2 Event Co-reference

DELTA 1.0 did not include any relation to co-refer events in a document. Annotators were forced to mark such relations as `SIMULTANEOUS`. One problem this caused was the disagreement in defining the `HET` anchor. Consider this example from `Document #AP900816-0139`:

*"We continue to **pray** and **pray** hard to God so that there will be no confrontation whereby you will **receive** thousands of Americans wrapped in sad coffins after you had pushed them into a dark tunnel".*

Annotators disagreed on which of the 2 *"pray"* events should be the anchor for the hypothetical timeline marked by the closest verb *"receive"*. One annotator assumed it should be the first occurrence, while the other assumed it should be the last/latest occurrence. Due to this disagreement, the relation had to be eventually marked as `VAGUE`. With the introduction of `EQUAL` relations, annotators could define either relation as an `HET`, and relation between the co-referring event and the closest event on the hypothetical timeline was automatically inferred as `HET` as well.

In some other cases, the 2 co-referring events themselves were marked on different timelines, leading to a chain of `VAGUE` relations.

We solve all these problems using a multi-step approach to document annotation, which along with a robust process and an updated, simplified annotation tool, achieves a high IAA score.

3.3 DELTA 2.0 - Annotation Schema

As discussed earlier, there are multiple steps associated with making an accurate event relations dataset. Our scheme consists of multiple layers of annotation which are described below:

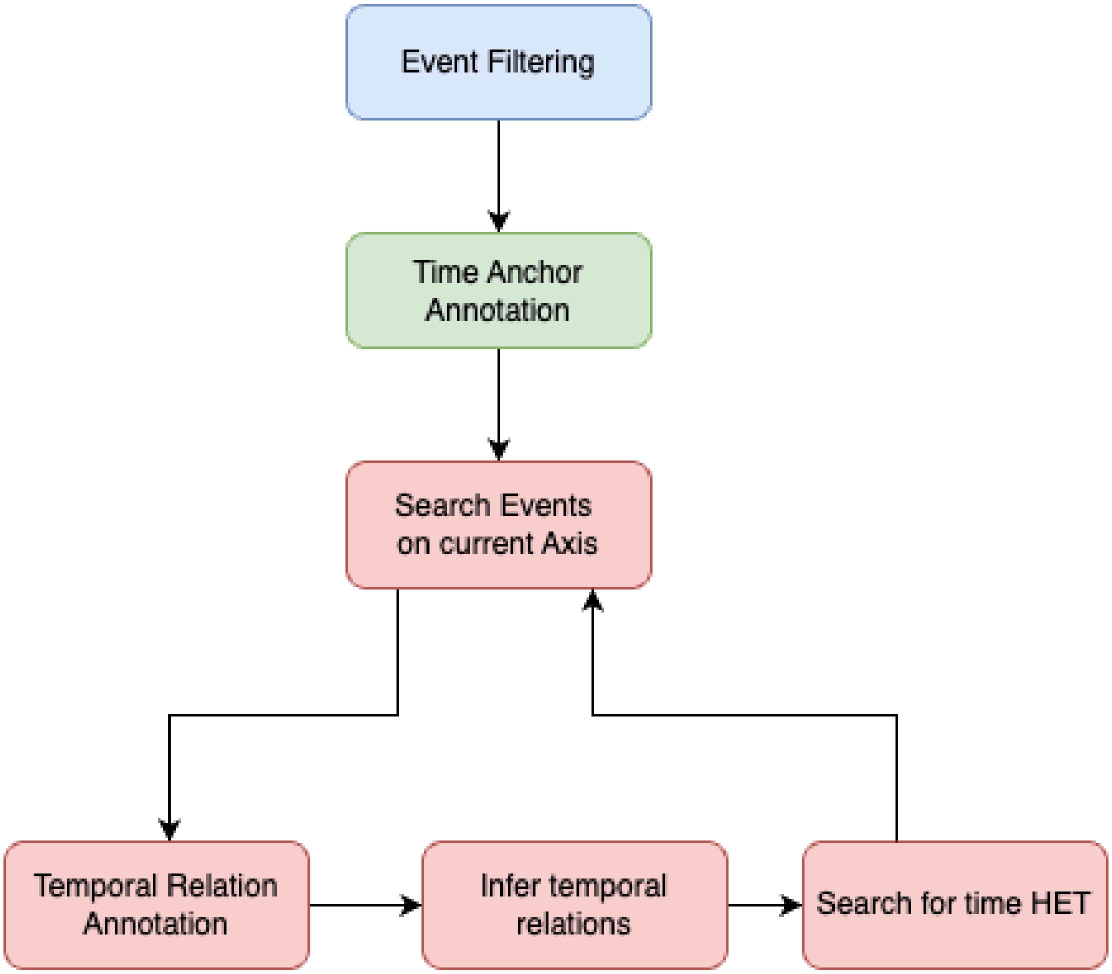


Figure 3.1: DELTA 2.0 annotation methodology. A series of iterations are required for the steps in red.

3.3.1 Event annotation(filtering)

Events in our corpus were annotated according to the TimeML guidelines (Saur 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 verbal phrases like, ”said”, ”kicked” or nominal events like ”war” or ”demonstration”. For the purpose of this dataset, we focus only on verbal events. In addition to this, unlike other datasets that ignore negated or hypothetical events, we include these in the dataset. However, in comparison to DELTA 1.0, we do not assume negated events to be the same as positive events. In some contexts, negative events are part of the main timeline, while in other, they are part of a hypothetical timeline. For example, in “Jack did not go to the class“ the event ’go’ does not actually happen. However, in “Mr.

Ronald cancelled the class", the act of cancelling happens in the real timeline. This distinction is important is determining which events to keep in the current timeline and which ones to keep in a parallel hypothetical timelines as both might have causation in their own temporal spaces.

3.3.2 Time Anchor Annotation

In order to improve the accuracy of temporal relation annotation, we used the notion of narrative container (NC), taking inspiration from earlier research (Pustejovsky and Stubbs, 2011). When an event without a clear time anchor occurs, NC, the default interval around the document creation time (DCT) of an article, estimates when it happened. It is influenced by several text style and genre-related factors; for instance, the NC value for newspapers is 24 hours, whereas it is a week for weekly publications and a month for monthly publications. Our annotators were informed that we may set the value of the narrative container to 24 hours because the newspaper articles in our corpus are published every day. Moreover, the DCT for each news item was given to annotators. If external and background knowledge aids in the annotation of more precise temporal anchors, annotators were encouraged to use it. When an event happened across a period of time, commentators were requested to supply the time anchor based on the interval's beginning. Events in our annotation approach are always assigned a date, either explicitly or implicitly. Annotators were specifically instructed to select one of six possibilities based on the type of temporal information connected with each event in order to enter the time anchor of the form YYYY-MM-DD. For example, the annotator notes "2024-04-14" if the temporal information related to the event is stated explicitly in the text (e.g., "April 14, 2024"). The following guidelines had to be followed in order to set time anchors for events:

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.

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".

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.

4. If the event happens in the past or in the future, with some idea about the date, the annotator can add that date. For example, for ”next Tuesday”, the annotator can add the date of next Tuesday after DCT.

5. 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 “XXXXXX-XX”. Figure 4 shows how the events are represented in a timeline.

6. If the annotator understands from the text that the event does not happen in the real world, he/she can mark it as ”HHHHHH-HH”

3.3.3 Searching events on the Axis

Once time anchors are assigned to events, we can choose the events on the current timeline/axis. We start with the main axis ofcourse, containing events which were annotated with a non HHHHHH-HH. This is important, as this makes the first step really simply for annotators and they can already begin to visualize the timeline. Using the annotation tool, annotators can click on the events they want to include in the current timeline. To make it easier for users to pick events on the current axis, we have them answer the following questions.

1. 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 already selected?

2. Is this event with an unknown date? If so, did it happen before or after another event in the same sentence?

This process of picking events makes the next step i.e. annotating event relations much easier and is directly in line with our idea of localizing events to multiple discourse timelines. We assume that events that have an Equal relation between them, must occur on the same axis. Hence, we mark ’equal’ relations in this step by asking the following questions:

1. Does the event refer to another event among the selected events?

2. Did the event start or happen at the same time when another event in the same sentence happened?

3.3.4 Temporal Relation Annotation and Automatic Inference

Unlike DELTA 1.0, where annotators had to re-read the entire document every time to find the relation between event pairs, since we’re working with a limited set of events, with time anchors, at a time; using the rules defined in Section 3.2, it is fairly straightforward to define BEFORE, SIMULTANEOUS, or DURING relations between events. The annotation tool helps here by picking all adjacent events first and prompting the user to annotate the relation between the 2 events. Once all relations have been marked, it runs automatic inference based on simple rules:

1. Transitivity: If there are three events, E_1, E_2, E_3 and the relations E_1 `before` E_2, and E_2 `before` E_3 exist, then the relation between E_1 and E_3 can automatically be inferred as `before`.

2. Temporal Equivalence: If there exists two events:E_1 , E_2, and there exists a relation E_1 is `simultaneous` with E_2 , then E_1 and E_2 will share the same relation with all other events. That is, if there exists another event E_3, such that the relation between E_1 and E_3 is `r`, then the relation between E_2 and E_3 is also `r`.

3. Temporal Equivalence: If there exists two events:E_1 , E_2, and one of them has a time-anchor XXXXXX-XX, all relations from and to that node will be marked as No Relation and this will be treated like an orphan event for the rest of the annotation process.

Now, in case there are any missing links, the annotation tool does another pass with events that are 1 hop away. For example, if e1, e2, and e3 occur in the text in this order, and there's a missing relation between e1 and e3, the tool will prompt the user to define a relation between the two. The hop count is increased until no more relations are possible. We noticed that because of defining the axis early and resolving coreferent events, the average hops for documents turns out to be 2, which is fairly low. Fig: 3.2 shows when the relation between 2 events can or cannot be automatically inferred. The links in red mark the inferred relations.

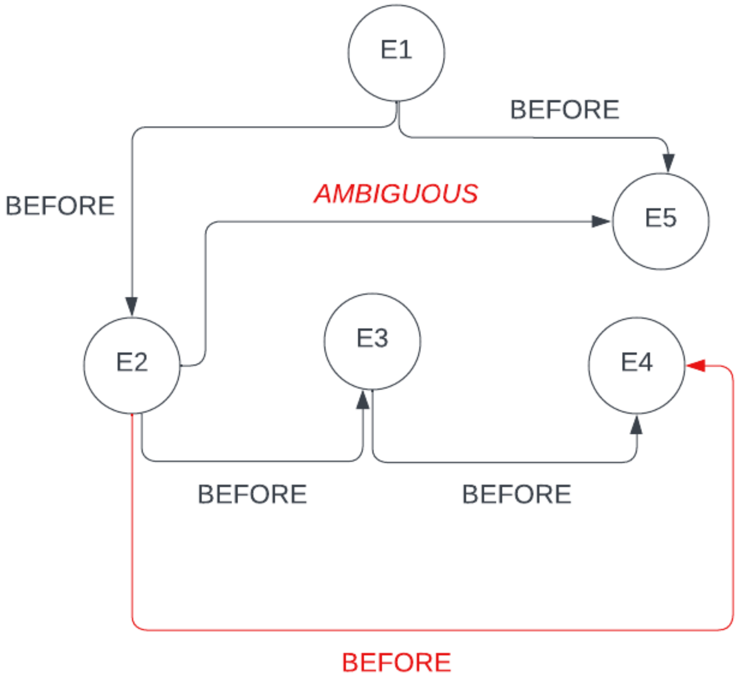


Figure 3.2: Automatic Relation Inference.

3.3.5 Search for HET anchor

After marking all events on a particular axis, annotators are nudged to search for new hypothetical axis. This is done by looking at the corpus and identifying events that can act as anchor to parallel axes. For instance, in the sentence ”Dean thought that kicking the ball hard won’t matter as the German goalkeeper would save it nonetheles.”, the event ”thought” serves as an anchor to a hypothetical timeline, for the relation between the events ”kicking” and ”save” to lie on the same lexically relevant timeline. Based on the learnings from DELTA 1.0, we were able to compile a list of rules ti identify possible HET anchor:

1. Verbs following modal verbs like ”could,” ”would,” ”should,” ”might” - These verbs often suggest possibility or speculation, which can be indicative of hypothetical situations.

2. Performative verbs like ”imagined,” ”thought,” ”dreamt,” ”supposed” ”wondered,” ”hoped” - These verbs introduce hypothetical scenarios or thought processes.

3. Events following Conditional clauses - Sentences starting with ”if” often introduce hypothetical scenarios based on certain conditions.
4. Adverbs like ”perhaps,” ”maybe,” ”possibly,” ”hypothetically” - These adverbs introduce uncertainty or hypothetical contexts.

Annotators are given these candidate events, and asked a couple of questions to determine if the event is an HET:

1. Are there any events that happen that can be linked directly to the current event?
2. Do the linked events exist on a non-primary axis?

If the answer to both the questions is yes, annotators can create a new timeline and restart the process described in Section 3.3.3.

3.4 Annotation Tool

A significant hurdle in creating high-quality datasets for temporal relation extraction (TRE) lies in the low Inter-Annotator Agreement (IAA) scores often observed. This can be attributed, in part, to the limitations of existing annotation tools when it comes to visualizing event relations and timelines. Current tools like Prodigy[1] and GraphAnno[2], while valuable for various tasks, fall short when dealing with TRE on longer texts. Annotating temporal relations within lengthy passages using these tools often requires frequent context switches, which can be cognitively taxing and impede efficiency. Additionally, representing temporal data as a simple ”triple” (event, event, label) makes grasping the intricate relationships between events a significant challenge. This lack of visual representation hinders annotators’ ability to fully comprehend the temporal structure within a document. Imagine trying to understand the complex flow of events in a news article or historical narrative by solely relying on a list of triplets! To address these limitations, we propose the development of a specialized annotation tool for TRE. This tool would not only streamline the annotation process by providing a user-friendly interface for marking event relations, but also offer crucial visualization capabilities. By visualising these relations as a graph or timeline, annotators can gain a clear understanding of the temporal structure within the text, leading to improved consistency and accuracy in their annotations. This visualization component would not only benefit the initial annotation process but also be instrumental in subsequent stages. By allowing reviewers and researchers to readily visualize the annotated timelines, the tool would facilitate collaboration, error identification, and overall comprehension of the temporal relationships within the data. A screengrab of the tool is shown in Fig: 3.3

[1] https://prodi.gy
[2] https://github.com/LBierkandt/graph-anno

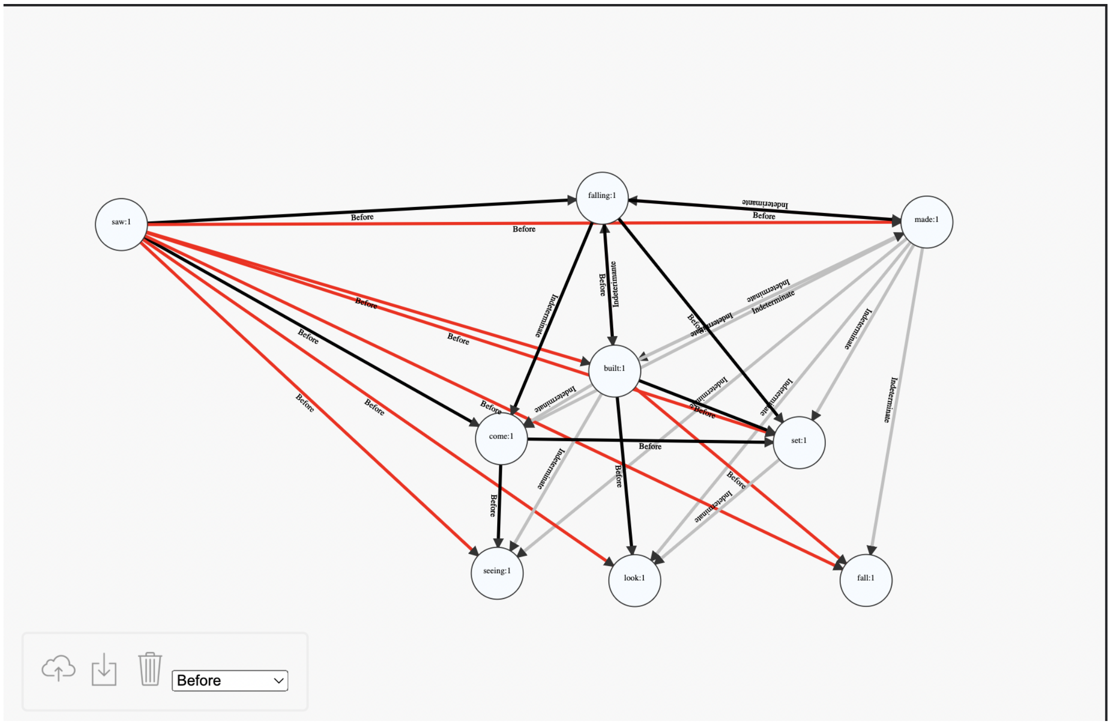


Figure 3.3: DELTA annotation tool.

Utilizing the tool simplifies the annotation process to the extent of drawing a connection between two event nodes. We've deliberately limited the visualization to events within the document to facilitate seamless annotation of cross-document relations. Annotators also possess the option to deduce new relations from existing ones. The edges are distinguished by color, denoting the type of relation inferred and enabling annotators to validate its accuracy. In instances where inference introduces ambiguity, potentially leading to two relations, annotators are prompted to specify the correct one.

3.5 Dataset Statistics

We conduct annotations on TimeBank-Dense, consisting of 36 English news documents. Other datasets like MATRES also use the same set of documents. The number of relations in each document ranges from as low as 32 to as high as approximately 17,600, with an average of 1,200 event-event temporal relations per file. Leveraging existing TimeBank annotations, we identify events within the corpus while filtering out all nominal events. A breakdown of the relations by class is provided in Table 3.1. Our annotations result in a gain of approximately 150 times more relations than MATRES, allowing for the capture of more long-distance relations. Additionally, inferring new relations from existing ones significantly aids annotators, resulting in an increase in the number of relations by nearly 185 times. However, due to their inherent nature, the relations categorized as `indeterminate` and `vague` contribute less to the timeline and are significantly more frequent than the other four relations. Therefore, we group these relations together as *frequent relations*, distinguishing them from the remaining four relations, namely `before`, `during`, `simultaneous`, and `HET`, which we classify as *infrequent relations.*

Using the new annotation schema, we also halved the number of VAGUE relations as compared to DELTA 1.0, which is a significant reduction. The automated relation inference based on the manually annotated increases the total number of relations by ∼185 times. While most of the inferred relations belong to the *frequent relations* set, the number of *infrequent relations* also rises substantially with inference, leading to $\sim 2.5X$ gain in the number of relations. We report a Kappa score of **0.82** as compared to 0.71 in DELTA 1.0 showing substantial agreement between the annotators.

Label	DELTA 2.0	DELTA 1.0
BEFORE	2178	2038
HET	182	202
DURING	65	67
SIMULTANEOUS	807	807
EQUAL	62	NA
VAGUE	11061	17095

Table 3.1: Class wise distribution of event relations in DELTA. Notice the difference in the number of VAGUE relations between the 2 datasets.

3.6 Baseline Models

This work explores the effectiveness of RoBERTa-based models ([Liu et al. (2019)]) for identifying temporal relations between events in text. We build upon the context encoder architecture proposed by Zhao et al. (2020b) with a key modification: expanding the context window beyond local sentence boundaries.

Problem Formalization

We represent a document (doc) as a sequence of sentences (s_i). Each event pair (p_i) consists of two event mentions (e_{i1} and e_{i2}) potentially located in different sentences (s_{ia} and s_{ib}). The goal is to predict the temporal relation ($y_{(i1,i2)}$) between these events (before, after, simultaneous, hypothetical, equal).

Baseline Model and Experimental Setup:
Local context features are generated for each event pair using a window of neighboring sentences (nbr_l before and nbr_r after the event mentions). Without loss of generality, let $ia \leq ib$.

A language model (LM) takes the concatenated sequence of sentences (seq_i) for each event pair as input. This sequence encompasses event mentions and potentially surrounding context based on the expanded window size. The LM generates contextualized embeddings (h_{i1} and h_{i2}) for each event mention within the sequence. A feature vector ($h_{(i1,i2)}$) is constructed by concatenating various elements:

1. Individual event embeddings (h_{i1} and h_{i2})

2. Element-wise absolute difference between embeddings ($|h_{i1} - h_{i2}|$)

3. Hadamard product of embeddings ($h_{i1} \circ h_{i2}$)

A classifier (C) takes the feature vector ($h_{(i1,i2)}$) as input and predicts the temporal relation ($y_{(i1,i2)}$). We evaluate the model on a dataset of 36 documents with an 80-20 train-test split. The baseline model has approximately 123 million parameters, primarily consisting of the pre-trained RoBERTa base model with a small classifier head. Training is conducted on an Nvidia 2080Ti GPU with each epoch taking around 30 minutes. We run experiments for 20 epochs with a batch size of 32 and a learning rate of 2e-5.

The results of our experiments have been reported in Tab.3.2. Compared to the same model applied to the TDDiscourse dataset (known for its focus on local sentence context), we observed a significant 48% increase in F1 score. This suggests that capturing long-range dependencies through our expanded context window approach plays a crucial role in accurate relation extraction. Despite the improvement, our baseline model still exhibits limitations in capturing all long-term dependencies within documents. This aligns with observations made on models trained on TDDiscourse, indicating a persistent challenge in this area. One encouraging finding is the improved performance on predicting hypothetical events. This can be attributed to the clear distinction we made within our annotation scheme between hypothetical and factual timelines. This distinction provides the model with valuable information that previous models, lacking such differentiation, might struggle with. This further strengthens the argument for incorporating the concept of multiple timelines into temporal relation extraction models.

Label	DELTA 1.0			DELTA 2.0		
	P	R	F1	P	R	F1
BEFORE	28.2	18.8	22.6	45.4	36.8	40.7
HET	99.0	80.0	88.9	99.0	81.2	89.2
DURING	0.0	0.0	0.0	0.0	0.0	0.0
SIMULTANEOUS	68.3	32.2	43.8	52.0	42.4	46.7
INDETERMINATE	49.9	90.8	64.4	NA	NA	NA
EQUAL	NA	NA	NA	80.0	48.6	60.5
Overall	49.3	49.2	49.3	57.0	54.8	55.9

Table 3.2: Evaluation metrics for the model on DELTA1.0 and DELTA 2.0. We report the metrics for non-vague event relations to avoid any bias caused by the number of *VAGUE* relations

3.7 Timeline Evaluation

To assess the quality of timelines generated by our DELTA model, we conducted a human evaluation study. Here's a detailed breakdown of the process:

Document Selection:
A representative sample of 10 documents was chosen from the corpus for evaluation.

Timeline Construction from External Datasets:
We constructed timelines for these documents using two existing datasets: MATRES and TDD. These timelines were created by adapting our inference rules to their specific label formats and automatically inferring relations wherever possible.

Minimal Timeline Creation:
All three timelines (DELTA, MATRES, and TDD) were then transformed into minimal timelines. A minimal timeline, in our context, refers to a timeline with the least number of relations while still guaranteeing that all possible relations can be inferred from this minimal set. This step helps ensure a fair comparison by focusing on the core relations in each timeline.

Human Evaluation:
Six human volunteers participated in the study. Each volunteer was presented with the three timelines for each of the 10 documents. They were asked to evaluate the timelines based on two key criteria:

1. Coverage: This criterion assessed how well each timeline captured the overall temporal structure and relationships within the document.

2. Accuracy: This criterion measured the correctness of the relations represented in the timeline, focusing on whether the relationships were factually accurate based on the content of the document.

Results:
The evaluation revealed a strong preference for DELTA timelines among the volunteers:

1. Coverage: In 68.3% of the cases, evaluators considered DELTA to have the best coverage of the temporal relationships within the document.

2. Accuracy: DELTA timelines were chosen for superior accuracy in 58.3% of the evaluations.

3. Overall Preference: Across all documents, 59.1% of the volunteers favored DELTA timelines over those generated from MATRES and TDD.

Chapter 4

Generating Event Timelines

4.1 Introduction

Temporal Relation Extraction (TRE) constitutes a fundamental aspect of Natural Language Processing (NLP), holding significance in diverse applications such as question answering and information retrieval. In recent years, there has been a notable surge in research efforts aimed at automating the inference of event relations, particularly temporal relations among events. Our study seeks to delve into the landscape of TRE by comprehensively evaluating existing methodologies to gain a deeper insight into the underlying mechanisms of each constituent model.

Employing a reproducibility methodology, we embarked on an extensive analysis encompassing various models designed to tackle the TRE task. Noteworthy among these are TIMERS Mathur et al. (2021), CTRL-PG Keskar et al. (2019), DEER Han et al. (2020), as well as certain non-neural, rule-based approaches. Among the myriad options, our focus primarily gravitated towards TIMERS, recognized as the state-of-the-art model in this domain. Notably, TIMERS employs a relational graph-based convolutional network architecture to facilitate the prediction of event relations, positioning it as a promising candidate for our investigation.

A key motivation behind our selection of TIMERS lies in its novelty and the availability of results on prominent datasets such as MATRES, TDD-Man, and TDD-Auto. Leveraging these datasets not only enables a fair comparison of TIMERS' performance against existing benchmarks but also facilitates extrapolation of its potential performance on the DELTA dataset. Additionally, the absence of an open-source version of TIMERS prompted our endeavor to develop an adapted version, which we have duly christened Adapted-TIMERS. By releasing this open-source iteration, we aim to foster greater collaboration and advancement within the TRE research community, thereby fostering further innovation in this domain.

The TIMERS paper proposes a novel approach using GRGCNs to incorporate document-level information, achieving significant improvements over existing models, achieving state-of-the-art results on multiple datasets. Reproducing these findings strengthens the validity of the proposed approach and its potential impact on TRE. Replication allows other researchers to verify the original findings and build upon them. It fosters trust in the research and allows for further exploration and refinement of the proposed techniques. Unfortunately, the original paper does not provide an open-source version of the code. This reproducibility study aims to address this gap by providing an open-source implementation, enabling further research and development in this area.

4.2 Reproducibility Study

4.2.1 Summary

For this reproducibility study, we first focus on developing an open–source implementation of TIMERS as it was not made publicly available by the authors. Then, we focus on verifying the two main claims made in Mathur et al. (2021): (1) TIMERS outperforms previous methods by a performance increase of 5.1-8.8 points on the TDDiscourse Naik et al. (2019), TimeBankDense Cassidy et al. (2014), and MATRES Ning et al. (2018b) datasets due to document-level modeling, and, (2) TIMERS shows improvement for event pairs that require chain reasoning, causal prerequisite links, and future events. Finally, we run ablations to figure out which of the new features added on top of the contextual embeddings add most of the value.

4.2.2 Task Defintion and Datasets

The task of Temporal Relation Extraction (TRE) involves the computation of the temporal relationship between pair of event mentions (e_{i1},e_{i2}) in a document(d_i). Each event pair is classified into one of the classes encoding a temporal relationship, e.g. BEFORE, AFTER, EQUAL, VAGUE in the MATRES annotation (Ning et al., 2018b); and BEFORE, AFTER, INCLUDES, IS INCLUDED, SIMULTANEOUS, AND VAGUE in the Timebank-Dense annotation. The scope of the task in TIMERS is to predict relations between a pre-annotated set of event–pairs between a document.

We replicate results on all the four datasets used in Mathur et al. (2021). MATRES Ning et al. (2018b) and TB-Dense Cassidy et al. (2014) classify temporal relations between events that are either in the same or consecutive sentences. While TB-Dense considers all possible such pairs, MATRES further restricts events on the same semantic axis. The other two datasets, TDDMan Naik et al. (2019) and TDDAuto Naik et al. (2019), do not put any restrictions and may contain event pairs from different parts of the document. However, they may not consider all possible event pairs, resulting in sparse annotations.

4.2.3 Model description

The TIMERS paper by Mathur et al. (2021) proposes a novel approach for Temporal Relation Extraction (TRE) that leverages document-level information to improve performance. This section delves into the key components of the TIMERS architecture and explains their contribution to the overall system.

1. Context Encoder (CE): Extracting Local Semantics The foundation of TIMERS lies in a ”context encoder,” which is essentially a fine-tuned Large Language Model (LLM) such as BERT. This fine-tuning process allows the LLM to specialize in the task of TRE. When presented with a document containing an event pair, the context encoder generates contextual embeddings (denoted as O_CE) for each event mention. These embeddings capture the semantic meaning of the event within the immediate context of the surrounding sentence(s).

2. Unveiling Document-Level Information: EDUs and Time-Aware Embeddings

   While the context encoder focuses on local semantics, TIMERS goes beyond this by incorporating document-level information. The paper proposes three additional features that capture this broader context:

   (a) Syntactic Embeddings (O_{DG}): The syntactic relationship between words is represented as an edge between them. Pre–trained BERTDevlin et al. (2018) is used to initialize all the node embeddings and message passing on $\mathcal{G}_{SG}$ using GR-GCN creates document–aware embeddings of each token, sentence and document.

   (b) Rhetoric Embeddings (O_{EDU}): To capture the discourse structure of the document, TIMERS constructs a ”discourse graph” (G_{DG}). This graph utilizes Elementary Discourse Units (EDUs) as nodes. EDUs represent minimal discourse units that convey a coherent proposition. The edges in the graph represent rhetorical relations between EDUs, defined by Rhetoric Structure Theory (RST). By employing message passing on G_{DG}, TIMERS extracts ”rhetoric embeddings” (O_{EDU}) that encode the discourse-level context of each event mention.

   (c) : Time-Aware Embeddings (O_T): This feature captures temporal information within the document. TIMERS constructs a ”time-aware graph” (G_TG) that models two key aspects: Temporal relationships between time expressions (TIMEX) and document creation time (DCT): The graph encodes the temporal ordering of explicitly mentioned time expressions within the document relative to the document’s creation time. Temporal arguments between events and TIMEX: A pre-trained Semantic Role Labeling (SRL) model identifies temporal arguments linking events to time expressions. These relationships are also incorporated into the graph. Message passing on G_{TG}

generates ”time-aware embeddings” (O_T) that capture the temporal context surrounding each event mention.

3. : Putting it Together: Classification with MLP All three features – O_CE (contextual embeddings), O_EDU (rhetoric embeddings), and O_T (time-aware embeddings) – are then concatenated into a single vector. This combined vector serves as the input to a Multi-Layer Perceptron (MLP) classifier. The MLP is trained to classify the temporal relationship between the event pair based on the rich contextual information captured by the concatenated features.

In summary, TIMERS leverages a combination of a fine-tuned LLM for local semantics (context encoder), discourse analysis for document structure (rhetoric embeddings), and a time-aware graph for temporal context (time-aware embeddings). By feeding this comprehensive set of features into an MLP classifier, TIMERS aims to achieve superior performance in TRE tasks compared to models that rely solely on local context.

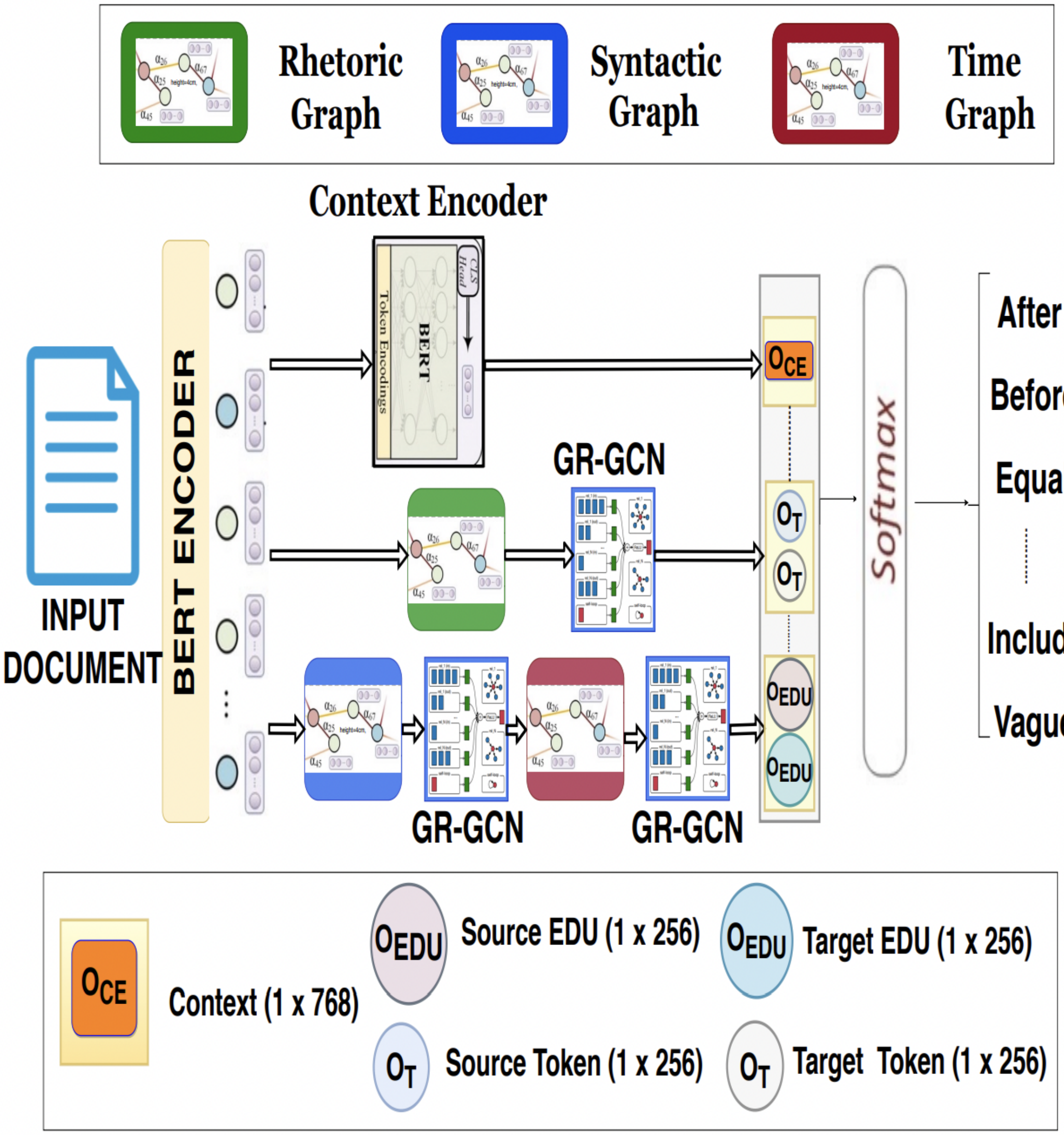


Figure 4.1: TIMERS model architecture. The 4 main components: CE(Context Encoder), Syntactic Graph, Time Graph, and Rhetoric Graph with input and output vector dimensions are shown in the figure.

4.2.4 Parameters and Requirements

The authors provide the optimal hyperparameters in the paper, and we use the same for all our experiments. Wherever possible, we use the same batch size as mentioned in the paper, but in some cases, we reduce the batch size to fit the model in the available GPU memory.

According to the paper, Mathur et al. (2021) ran all experiments on a single 8GB Nvidia GeForce RTX 2080 GPU. However, we found that 8 GB GPU memory is insufficient, especially for TDDAuto dataset, where the largest document has 432 event pairs. Further correspondence with the authors revealed that they indeed used multiple GPUs to fit the mentioned batch size in the memory. Due to non–availability of multiple GPUs, we ran all our experiments on a single node with two RTX 5000 with 16GB GPU memory.

4.2.5 Analysis

4.2.5.1 Results from study

Section 4.2.5.1 shows a comparison between the F1 scores reported in Mathur et al. (2021) (F1–reported) and the scores obtained in our work (F1–ours). Below we discuss each of their central claims in detail.

	F1–orig				**F1–rep**				**F1-orig**	**F1-rep**
	TDM	**TDA**	**MAT**	**TBD**	**TDM**	**TDA**	**MAT**	**TBD**	**Average**	
Baseline (CE)	37.5	62.3	77.2	62.2	36.6	60.5	77.8	61.6	59.8	59.1
TIMERS	45.5	71.1	82.3	67.8	41.8	64.8	80.0	63.8	66.7	62.6
Gains	8.0	8.8	5.1	5.6	5.2	4.3	2.2	2.2	6.9	3.5
	Drop in performance of TIMERS by excluding a component.									
w/o CE	11.8	19.5	13.7	17.2	13.4	18.2	18.7	16.7	15.6	16.8
w/o $\mathcal{G}_{DG}$	3.7	5.7	2.6	4.5	3.4	2.9	1.6	1.3	4.1	2.3
w/o $\mathcal{G}_{SG}$	3.2	2.2	4.1	5.0	3.6	2.4	0.8	2.1	3.6	2.2
w/o $\mathcal{G}_{TG}$	6.0	4.0	4.6	6.0	1.6	4.8	1.0	1.7	5.2	2.3

Table 4.1: A comparison of the F1 scores reported in the original paper against the scores obtained in our experiments. We focus on comparing the gains achieved by TIMERS over the baseline. For ablations, to understand the importance of each component, we compare the drop in TIMER's performance by removing the component. We report metrics over individual datasets as well as averages over all of them. *TDM: TDDManual ; TDA: TDDAuto ; MAT: MATRES ; TBD : TBDense*

The importance of each component in resolving relations with particular discourse properties has been analyzed in the original work. Based on the design of each component:

- The Syntactic-aware graph ($\mathcal{G}_{SG}$) is important for temporal relations that can be extracted from a single sentence (SE)

- The time-aware graph ($\mathcal{G}_{TG}$) is important for determining temporal relations that require multi–hop chain reasoning (CR) and are determined by future events (FE).

- The rhetorical-aware graph ($\mathcal{G}_{DG}$) is important for event pairs whose temporal relationship depends on Future Events (FE).

We first observe that incorporating document–level features, as proposed in TIMERS, indeed results in gains over the baseline Context Encoder. In particular, for TDDMan and TDDAuto, which contain long–range event pairs, we observe significant gains. As expected, the gains for MATRES and TB–Dense are much less, as they restrict event pairs within consecutive sentences and hence may not benefit from using document–level features. However, we also observe that the gains are not as high as reported in Mathur et al. (2021): their gains range between 5.1 pts (MATRES) and 8.8 pts (TDDAuto), whereas we observe gains between 2.2 pts(MATRES) and 5.2 pts (TDDMan).

To measure the relative importance of each component, we focus on the drop in TIMERS performance on removing a particular component from it. We first observe that the performance drops the most by removing the baseline context encoder (CE) from the model, demonstrating that the baseline CE is still the most significant component.

On average, we observe a similar drop (2.3 pts) in performance on removing any of the three proposed components from TIMERS. On the other hand, the original paper reports a maximum drop by removing the time graph $\mathcal{G}_{TG}$(5.2 pts), followed by the discourse graph $\mathcal{G}_{DG}$(4.1 pts), and syntactic graph $\mathcal{G}_{SG}$(3.6 pts). When averaged across all datasets and components (4.3 and 2.3 for the reported and ours, respectively) and measured as a fraction of the average gain over the baseline (6.9 and 3.5, respectively), the drop observed in our experiments (2.3/3.5=65%) is similar to what is reported in the original work (4.3/6.9=62.5%). In our experiments, the most significant drop is observed in TDDAuto (4.8 pts) on removing $\mathcal{G}_{TG}$. This is expected as TDDAuto may require multi–hop reasoning due to the presence of long–range event pairs, unlike MATRES and TB-Dense. On the other hand, unlike Mathur et al. (2021), we do not observe much difference on TDDMan.

	Original					**Reproduced**				
Phenomena	**SS**	**CR**	**FE**	**HN**	**CP**	**SS**	**CR**	**FE**	**HN**	**CP**
count	100	10	32	8	32	100	10	32	8	32
TIMERS	35	38	39	35	55	41	30	37	25	47
w/o $\mathcal{G}_{DG}$	25	30	16	18	15	40	20	21	12	15
w/o $\mathcal{G}_{SG}$	1	23	32	42	48	33	20	28	25	34
w/o $\mathcal{G}_{TG}$	22	0	8	37	52	35	10	25	25	34

Table 4.2: Results comparing the percentage of relations resolved in different categories based on *relation phenomena*. SS: same sentence ; CR: Chain Reasoning; FE: Future Events; HN: Hypothetical/Negated; CP: Causal/Prereq

To do this analysis, we use the categorization of event pairs into one or more *relation phenomena* in TDDAuto dataset and measure the accuracy of the model separately for each of the categories. Table 4.2 compares the reported and observed accuracy in resolving relations corresponding to each phenomenon. It first reports the total count of event pairs in each of the categories. Apparently, reporting percentages over such small numbers may show huge differences even when there is a difference in only a few samples. Nevertheless, for the sake of consistency and comparison with the original work, we also report percentage accuracy for each category.

Importance of $\mathcal{G}_{SG}$ for SS: For event pairs within the same sentence, comparing TIMERS with the ablation without $\mathcal{G}_{SG}$, we do not observe as significant a difference (41 vs 33) as observed in the original paper (35 vs 1).

Importance of $\mathcal{G}_{TG}$ for CR and FE: Here also we observe similar trends as observed in the original work, albeit the drop in performance over CR and FE class on removing $\mathcal{G}_{TG}$ is not as high as reported in Mathur et al. (2021).

Importance of $\mathcal{G}_{DG}$ for FE, HN, and CP: Similar to our earlier observations, we see the biggest drop by removing $\mathcal{G}_{DG}$ for FE, HN and CP categories but the drop is not as significant as reported in the paper.

In summary, we were able to verify claims related to performance gains from overall model and individual components from the paper, however, the gains we observed were not as high as reported by the authors. Even after correspondence, we could not resolve the anomalies seen in relative and overall numbers, and we expect other researchers to also see similar numbers. For the claims on the importance of different components for different categories of event pairs, we observed similar trends as reported in the paper, but the quantum of differences is not as significant as reported in the paper.

4.2.5.2 Results beyond study

We also explore some modifications to the individual parts of the model, to see if any gains are introduced. Our observations are summarized below:

- **Can a different baseline context encoder give better results?**
  section 4.2.5.1 shows that the most significant component of TIMERS is the baseline Context Encoder, and hence we experiment with RobertA CE instead of BERT. While the baseline improves by 1.8 pts on average, however, there is no significant difference in the overall performance of TIMERS.

- **Can a different model like multi-heading GAT perform better than a RGCN?**
  We repeat our experiments by replacing GRGCN with Graph Attention Networks (GAT) in TIMERS, mainly because of ease of implementation and overall training time. However, we observe that it comes at the cost of overall performance.

4.2.5.3 Plausible reasons for discrepancies

1. **Differences in RST Discourse Parser:** $\mathcal{G}_{DG}$ graph leverages discourse information by computing relationships between segments of the document. These segments and relations are generated based on the Rhetorical Structure Theory UzZaman et al. (2013). Mathur et al. Mathur et al. (2021) use a private version of the library that extracts this metadata from the document. On the other hand, we use an open source library DMRST Parser to achieve the same. This may explain some of the discrepancies between the two implementations.

2. **Missing details for canonicalization of time–expressions:** The paper reports the usage of the Microsoft Recognizers-Text library to canonicalize time expressions in the documents. However, the exact details of how it is done are missing and we could not figure it out even after our correspondence with the authors.

3. **Insufficient computational resources for TDDMan and TDDAuto.** Mathur et al. (2021) use parallelization on multiple 8 GB GPUs to train the model on TDDMan and TDDAuto, however, this detail is missing in the paper. As mentioned earlier, we ran our experiments on a single 16 GB GPU, which restricted us to backpropagate on event pairs from only a single document in one iteration. Even though in a single update, the loss is computed over multiple event–pairs, all of them come from the same document, which may result in biased training. Nevertheless, this should not affect our claims, as we focus on relative gains over the baselines, rather than overall performance.

4.2.6 What was difficult

- **Unavailability of code:** The biggest constraint in this study was the unavailability of code. More specifically, we could not get details about implementations of the Rhetoric and Time Graphs, however, we managed to contact the authors and use alternative libraries for EDU prediction, discourse relation extraction, and generating edges in the time graph.

- **Computational constraints for TDDAuto and TDDMan:** We often found memory-related issues trying to run the model on datasets with dense annotation of event pairs in a document (TDD-Man and TDD-Auto).

- Even though we noticed comparable gains on including one graph at a time, we struggled to match the expected numbers on putting all 3 graphs together. We expect other researchers to also face similar issues.

4.2.7 Evaluating Timelines

We observe that precision, recall and F1 scores reported by Mathur et al. (2021) are not consistent with each other, *e.g.*, F1 score is greater than both precision and recall. Our conversation with the authors revealed that they used relaxed precision and recall, as suggested in Ning et al. (2018c), but the micro F1 score is computed in the usual way over all classes other than 'VAGUE'. In the case of the relaxed metric, all event pairs whose actual label is 'VAGUE' are assumed to be predicted correctly irrespective of the inferred label! Further, it is followed for only MATRES and TB-Dense datasets that have 'VAGUE' as one of the classes. For TDDMan and TDDAuto, which do not have a 'VAGUE' class, the metric is even further relaxed by assigning 'VAGUE' label to all event pairs that are not annotated in the dataset, resulting in 'VAGUE' being the most frequent class, and by the definition of relaxed metric, all predictions for this class are assumed to be correct.

Now with this relaxed definition of Precision and Recall, one should expect them to be much higher than usual micro-precision and recall. However, this is not the case with the reported results. Table 4.3 reports a comparison between the reported and observed relaxed precision and recall. For comparison, we also report the usual micro–precision and recall in the same table.

	Relaxed-Orig				Relaxed-Rep				Micro-Rep			
Dataset	**TDM**	**TDA**	**MAT**	**TBD**	**TDM**	**TDA**	**MAT**	**TBD**	**TDM**	**TDA**	**MAT**	**TBD**
						Precision						
CE	36.5	62.0	65.6	59.7	48.2	67.3	84.2	65	36.5	62.0	77.7	60.9
TIMERS	43.7	64.3	81.1	48.1	53.5	69.2	84	66.8	43.7	64.3	77.7	61.2
w/o CE	29.7	49.8	61.2	43.8	40.6	58.8	63.6	59.8	29.7	49.8	56.7	45.6
w/o $\mathcal{G}_{DG}$	39.6	61.7	71.8	51.4	50.3	66.6	84	63.2	39.6	61.7	77.5	60.9
w/o $\mathcal{G}_{SG}$	38.5	63.3	71.6	51.1	50.4	70.5	82.1	63	38.5	63.3	77.7	60.8
w/o $\mathcal{G}_{TG}$	37.5	58.7	72.8	50.5	49.6	65.2	84.2	63.2	37.5	58.7	77.6	60.8
						Recall						
CE	37.1	61.7	78.1	60.7	52.1	68.2	85.1	66.7	37.1	61.7	77.9	62.3
TIMERS	46.7	72.7	84.6	65.2	56.6	77.4	87.2	69.2	46.7	72.7	81.2	66.6
w/o CE	35.5	52.5	69.6	54.5	52	61.4	70	60.4	35.5	52.5	64.2	48.7
w/o $\mathcal{G}_{DG}$	39.6	66.8	79.1	63	53	73.6	86.1	67.1	39.6	66.8	80.3	64.2
w/o $\mathcal{G}_{SG}$	42.6	69.5	78.5	62.1	52.2	70	82.5	66	42.6	69.5	77.9	62.7
w/o $\mathcal{G}_{TG}$	39.8	68.3	78.5	62.9	53.1	75.4	84.8	67	39.8	68.3	80.1	63.5

Table 4.3: Results comparing relaxed precision and recall measures over all datasets TDM: TDDManual; TDA: TDDAuto; MAT: MATRES; TBD: TBDense

4.3 Using LLMs for predicting temporal relations

As big language models like ChatGPT get better, people wonder if they can be used to understand time and make timelines of events. We think even with these fancy models, it's still important to label events in detail to help train other models. Some researchers, like Yuan et al. and Alsayyahi et al., looked into how well these big models do at understanding time. Alsayyahi et al. found that models like ChatGPT don't do as well as supervised models at figuring out when events happen. They tested ChatGPT on tasks it hadn't seen before and found it didn't do as well as the trained models. In their study, they asked ChatGPT to find event times using a special method suggested by Yuan et al. ChatGPT didn't do as well as the trained models, getting only about 31% precision, 36% recall, and 33% overall accuracy. That's much lower than the trained models, which obtained 69.05% for precision, 69.05% for recall and 69.05% for F1-score for all those measures.

Some of the reasons for LLMs falling short on this tasks can be understood through the following challenges:

1. Limited Reasoning Ability: While LLMs excel at capturing statistical relationships between words, they struggle with tasks requiring explicit reasoning or logic. Determining temporal relations often requires understanding the flow of events and implied causality within a

sentence or document, which can be challenging for LLMs. Unlike seq-to-seq tasks where LLMs have seen a lot of success, TRE requires a deep semantic understanding of the entire document, which is hard for LLMs.

2. Focus on Local Context: LLMs primarily focus on the immediate context surrounding the event mentions. However, TRE often relies on broader discourse cues and relationships between events spread across sentences. The paper acknowledges that current LLMs may not inherently capture these long-range dependencies.

On the contrary, graph based model, like TIMERS, fair much better in this task. These models represent the document structure and relationships between events as a graph. Message passing algorithms are used to propagate information across the graph, potentially capturing long-range dependencies within the document for more accurate temporal relation extraction.

4.4 Adapted-TIMERS on DELTA

In this study, we introduced Adapted-TIMERS, an adapted, open-source version of the TIMERS model, and DELTA 2.0, a new dataset designed to explore multiple timelines. The evaluation of Adapted-TIMERS on DELTA 2.0 yielded promising results. 4.4 shows the results of the model on both versions of DELTA.

Label	DELTA 1.0			DELTA 2.0		
	P	R	F1	P	R	F1
Roberta Baseline	49.3	49.2	49.3	57.0	54.8	55.9
Adapted-TIMERS	51.2	52.0	51.5	61.4	59.2	60.3

Table 4.4: Evaluation metrics for the model on DELTA1.0 and DELTA 2.0. We report the metrics for non-VAGUE event relations to avoid any bias caused by the number of *VAGUE* relations

Notably, the dataset exhibited fewer non-vague relations, which contributed to an increase in Precision numbers. I've talked about the reason behind discarding VAGUE relations in detail in Section 4.2.7. This improvement underscores the effectiveness of Adapted-TIMERS in accurately capturing temporal relations, particularly in scenarios where vague relations are minimized. Furthermore, the performance of Adapted-TIMERS surpassed that of baseline models, demonstrating its superiority in temporal relation extraction tasks.

Chapter 5

Hindi TimeBank: Temporal Relations in Indic languages

The importance of building event temporal relations datasets for Hindi and other Indic languages has gained recognition in recent years. While Hindi has been added to the list of languages with literature focusing on event and temporal expression annotation, the existing efforts are still in their nascent stages. For instance, temporal expression identification in Hindi and basic classification have been addressed to some extent in the FIRE 2011 Palchowdhury et al. (2011) corpus. However, more targeted approaches have emerged, such as those proposed by Ramrakhiyani and Majumder in 2013 and 2015. These efforts highlight the growing recognition of the need for comprehensive datasets tailored to the linguistic nuances of Hindi.

In addition to temporal expression identification, efforts have been made towards event detection and recognition within the Hindi language framework. Goud et al. (2019) have established a framework and basic guidelines for binary recognition of event nuggets, distinguishing between events and states. This work underscores the complexity of event recognition within the semantico-syntactic grammatical framework of Hindi. It emphasizes the unique challenges posed by the linguistic structure and semantic nuances of the language, necessitating specialized approaches for accurate event identification and classification.

By focusing on the development of event temporal relations datasets for Hindi and other Indic languages, researchers can address the specific linguistic characteristics and cultural contexts inherent in these languages. Such datasets enable the training and evaluation of NLP models tailored to these languages, facilitating advancements in various applications including information extraction, question answering, summarization, and timeline generation. Ultimately, the availability of comprehensive datasets contributes to the broader goal of advancing natural language understanding and processing capabilities in multilingual and multicultural contexts.

We build on the idea and the first seed dataset presented in Goud et al. (2019)'s binary event categorization work in Hindi. Our addition includes classifying both events and states, as well as annotating states—a feature that previous studies purposefully omitted. We further complement the dataset by adding time expressions, which yields a complete set of 1,000 articles in Hindi. We offer comprehensive guidelines to help with correct identification and separation of events from states. For uniformity, the Hindi classification method for states and events follows TimeML guidelines. In addition, we incorporate language-specific tweaks and adjustments to the ISO-TimeML schema that are transferable to other languages. Lastly, we use inter-annotator agreement to assess how strong the annotation guidelines are. **Although this timebank dataset does not explore multiple timelines like DELTA, we've made considerable efforts to provide enough metadata for each event, state or time expression; as well as event relations; to enable researchers to easily adapt this to new annotation schemes. Out efforts will also assist in creating feature-rich vector representations when building models for automatic inference of event relations.**

5.1 Annotation Guidelines

TimeML, as defined by Saurí et al. (2006), characterizes events as situations that occur, hold, or take place, encompassing states or circumstances where something holds true. However, when applied to annotating Hindi events, annotators faced challenges due to low confidence levels. This stemmed from ambiguities in event normalization, subordinating verbs, and "generics," which were not to be marked. To address these challenges and enhance annotation accuracy, we introduce a refined definition of states. In this context, a state is characterized as a verbal predicate that offers a spatio-temporal description of participating entities, encompassing properties, location, or existence. This definition accommodates verbal modifiers and copular constructions, facilitating more precise and comprehensive annotation of Hindi events and states.

5.1.1 Events

Most of the event types are the same as those found in TimeML (Pustejovsky et al. (2003)). Thus, the following event categories are annotated:

1. **REP:** Reporting events entail the act of individuals or organizations declaring, narrating, or informing about an event.
   For example: *rAm ne **kahA** muJe BUka lagI hai* (Ram said that he is hungry).

2. **ASP:** Aspectual events encompass actions that signify the commencement, conclusion, continuation, or any other aspectual state of another event.
   For example: *bache ne KAnA **SurU kara xiya*** (The child started eating).

3. **PER:** This category comprises events that entail the physical perception of something by another entity.
   For example: *rAm ne muJe **xeKA TA*** (Ram had seen me).

4. **IAC:** Intensional Action events refer to events that explicitly introduce another event as an argument, distinct from serving as the aspectual state of that event. In Hindi, there are two syntactic forms of IAC events. They either manifest as the primary verb in the sentence, subordinating the other verb, or as the subordinating verb itself. In both instances, the I Action lacks completeness without another event or state.
   For example: *rAm kapade* ***pahanakara so gayA*** (Ram slept with his clothes on).

5. **OCC:** Any events not falling into the specific categories mentioned above are classified as occurrences, denoted by OCC. It's important to note that all nominal events are inherently considered occurrences.
   For example: *yuxXa meM sEnika* ***GAyala hue*** (Soldiers were hurt in the war.).
   Note that eventhough we saw that it's hard to annotate event relations for nominal events, we've included these in the timebank, so that annotation schemes in the future can utilize this information.

5.1.2 States

While TimeML includes an event category for STATE and I-STATE, our approach differs as outlined in Section 5.2. We do not classify states or intentional states as events. Therefore, we propose a schema for categorizing states based more on syntactic rather than semantic considerations. The introduced categories in the schema are declarative (DECL) and descriptive (DESC) states.

1. **DECL (Declarative):** A verb assumes the classification of a declarative state when it imparts information concerning the properties or attributes of a participating entity. These verbs are distinguished by their association with copular constructions, facilitating their unique identification.
   For example: *vaha gend lAl raga kI* ***hai*** (That ball is red in colour).

2. **DESC (Descriptive):** A verbal modifier or participle assumes the classification of a descriptive state when it can be reformulated as a copular construction and, in its capacity as a modifier, furnishes information concerning the entity or event it describes.
   For example: ***gAta huA*** *bacca kal bImAr thA* (The child who is singing was sick yesterday).

5.1.3 Time Expressions

Time expressions are defined as spans of text that represent specific times, durations of events or states, or points in time relative to other events or time expressions (Group and others, 2009). The annotation and evaluation of temporal annotations are essential concepts in information retrieval based on events, as events rely on time expressions as anchors. Therefore, the assessment of temporal annotations is widespread in semantic evaluation literature (Verhagen et al., 2010).

A time expression comprises several components: a unique ID known as "t id," which serves as a reference when the expressions act as anchors to TLINKs (explained in Section 3.5); a class, which can be DATE, TIME, DURATION, or SET; the tokens within the span of the time expression; and the AnnConf, representing the annotator's confidence parameter.

In Hindi, the classes of time expressions are described as follows:

1. **TIME:** The TIME category is designated for marking temporal references within text, encompassing specific instances like "7 o'clock" (sAt baje) as well as broader periods like "morning" (subah). It's important to include case markers or karakas associated with the time expression, especially when they convey durational information.
   For example: *rAm **aja sAt baje** aega (Ram will come at 7 o'clock today.*

2. **DATE:** The DATE category is designated for annotating specific calendar days, dates, weekdays, and other temporal expressions encompassing multiple days or dates, such as weeks, months, or years. It's important to note this label is used for point time, i.e., spans of time with defined start and end dates are not categorized under this label.
   For example: *rAm **do mahIne** bAd aega (Ram will come after 2 months.)*

3. **DURATION:** For non point-time constructions, the DURATION category is applied. This is used for marking spans of text with different start and end times in the document's context. For example: *rAm **sAt mahIno** se gayab hai (Ram has been missing for seven months.)*

4. **SET:** The SET class of time expressions serves to establish the periodicity of an action or denote an event occurring at a specific time in the past or future relative to the current time. Including the karaka is crucial as it indicates the duration or recurrence of the event.
   For example: ***hara cara sAla** olapiksa hote hai (The Olympics take place every four years.)*

5.1.4 Event Relations

This section goes over the various relations defined between events and states. These links provide important information about the relation between events and states and can be really helpful in downstream inference tasks as we've discussed before. We use a TimeML inspired annotation schema and categorize these linkages into 3 types. I'll focus most on the temporal link (TLINK) since it is the most important type of relation for generating timelines. However, it must be noted that the other 2 categories provide important metadata for future work.

TDDMan used similar discourse knowledge for generating their dataset. The categories and the annotation schemes are described below:

1. **TLINK:** Most events in natural language, have a start and an end time. We've seen in Section 3 how specific or approximate timestamps for events can be determined using a series of questions. Based on these timestamps, events are linked to other events; i.e., an event may occur before, with, or after another event. A TLINK is used to designate such relationships between 2 events, 2 states, or between an event and a state. We further divide TLINKs into 3 categories. Note that for all these we've assumed that the 2 events E_1 and E_2 can be anchored in time i.e. they have an inferable start and end time:

   (a) **BEFORE:** We say 2 events E_1 and E_2 have a BEFORE relation between them, if $finishTime(E_1) < startTime(E_2)$.
   For example: *rAm kamare meM **gayA** Ora **so gayA*** (Ram went to the room and slept).
   In the above example we're sure that the event gayA happened before the event so gayA.

   (b) **BEFORE-OVERLAP:** We say 2 events have a BEFORE-OVERLAP relation between them if, in the two events E_1 and E_2, $startTime(E_1) \leq startTime(E_2) < finishTime(E_1)$ and $finishTime(E_1) < finishTime(E_2)$.
   For example: *rAma **yuxXa** ke pahale se **gusse mE wA*** (Ram was angry even before the war).
   In the example above, the event gusse mE started before the event yuxXa, and didn't end before the event yuxXa finished.

   (c) **OVERLAP:** We say 2 events have have an OVERLAP relation between them, when, in the two events $E1$ and $E2$, $startTime(E2) \geq startTime(E1)$ and $finishTime(E2) \leq finishTime(E1)$.
   For example: *yuxXa meM rAm GAyala hue* (Ram got hurt in the war).
   In the above example the event GAyala happened during the event yuxXa.

2. **SLINK:** A subordination link, abbreviated as SLINK, is utilized to denote relationships between two events, typically between reporting events and other events. Additionally, we include certain intensional events in conjunction with other events, where the latter event either anticipates or dictates the former event. Conditional constructions are also annotated as SLINK.
   For example: *rAma **kahawA hE** kI yuxXa **gaMBIra hE*** (Ram says that the war is intense).
   In the example above, the event kahaWa hE is of reporting speech category, describing the event gaMBIra hE.

3. **ALINK:** An aspectual link, referred to as ALINK, indicates the connection between an aspectual event and its corresponding argument event or state. The ALINK tag encompasses four distinct classes: INITIATION, TERMINATION, CONTINUATION, and CONCLUSION.
   For example: *rAma ne kAna* ***Suru kara*** (Ram started eating)

5.2 Annotation Pipeline

Following up on Goud et al.'s (2019) work, where they proposed an 810 document dataset; we picked a subset of these documents, discarding all documents with the count of tokens less than 100. Such small documents did not have enough event relations to contribute to this dataset. A group of 8 annotators did multiple rounds of annotations in each step of the annotation pipeline. All annotators were native speakers of Hindi, and trained in linguistics. We follwowed the BRAT Annotation Framework (Stenetorp et al., 2012).

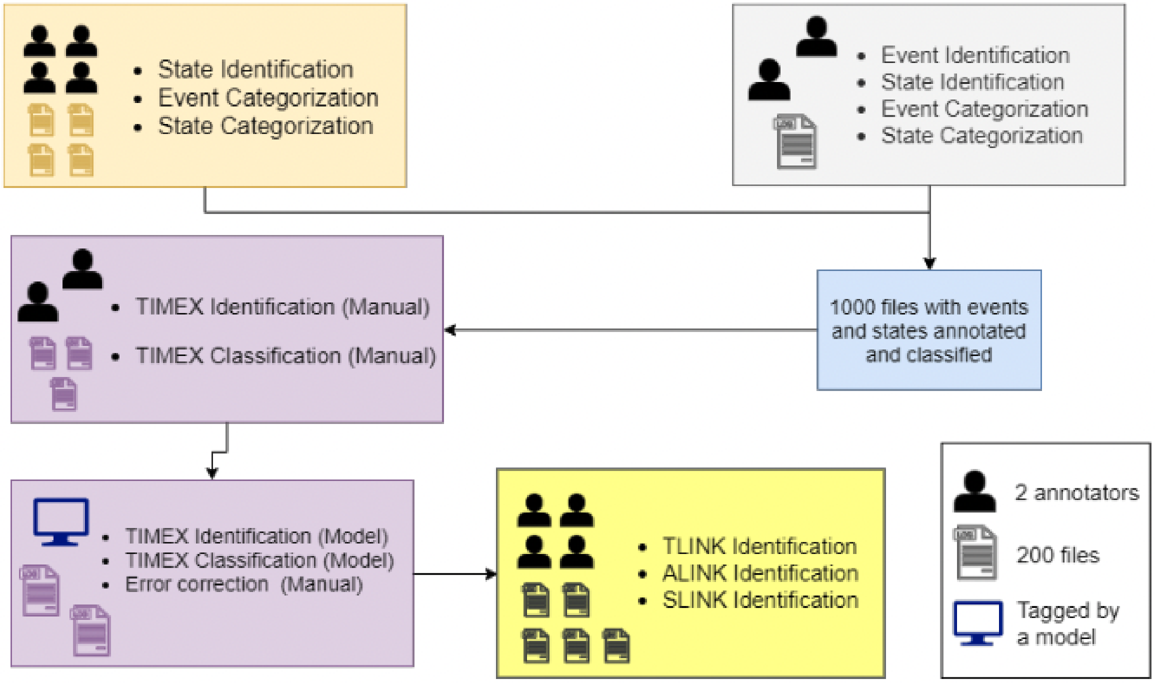


Figure 5.1: Annotation Sequence for Hindi-TimeBank. The legend for each icon used in the diagram is provided in the bottom right.

5.2.1 Event and State Identification

We've already talked about the source of the news articles for this dataset. As the dataset from Goud et al. (2019) exclusively comprised articles from the financial crime domain, it lacked balance in representing the various syntactic and semantic contexts in which events and states occur in Hindi. To address this imbalance, we augmented the seed dataset by incorporating 200 additional articles, consisting of 150 news articles sourced from Navbharat Times, a prominent national Hindi daily newspaper with a circulation exceeding 2 million copies. Additionally, we included 50 short fiction stories authored by Premchand, a celebrated Hindi writer. This diversification of article sources aims to provide a more comprehensive representation of the linguistic diversity present in Hindi text.

The distribution of the scraped articles is outlined in 5.1. To prepare these articles for annotation, we initially tokenized them using a freely available tokenizer (Reddy and Sharoff, 2011). Subsequently, the identification of both events and states within the articles was performed by four annotators in a series of four rounds, with each batch comprising 50 articles. During the annotation process, significant inter-annotator variability was observed, particularly in the identification of reporting verbs lacking a participating entity. Consequently, such constructions were excluded from consideration during event and state annotation to ensure consistency and accuracy in the dataset.

Article Domain	Article Count
Financial (Goud et al, 2019)	800
Fiction	50
National (News)	30
Business Analysis (News)	30
Entertainment (News)	30
Sports (News)	25
Technology and Development	25
Education (News)	100
Total	1000

Table 5.1: Domain-wise distribution of articles in Hindi Timebank

5.2.2 Event and State Classification

The set of 200 articles, previously introduced, underwent meticulous annotation for event and state categories by a team of four annotators. This annotation process unfolded in batches of 50 articles across four iterative rounds. The classification guidelines employed were predicated upon readily discernible syntactic disparities, thereby instilling a sense of high confidence among annotators during the manual annotation of events and state categories.

Subsequently, the dataset from Goud et al. (2019) comprising articles was subjected to a similar annotation procedure by a larger team of eight annotators. This annotation process unfolded in batches of 100 articles over three successive rounds. Consequently, this concerted effort culminated in the creation of a comprehensive corpus comprising 1000 articles, wherein event and state phrase boundaries were meticulously identified and classified.

5.2.3 TIMEX Identification

For identifying time expressions, we employed a Conditional Random Fields (CRF) model to tackle the task of identifying time expressions within the dataset. The CRF model was trained using a set of 600 articles that were manually annotated for time expressions. Subsequently, the model was evaluated on the remaining 400 articles to assess its performance.

The CRF model utilized four features viz. (1) Word Identity (WI), (2) Part Of Speech (POS), (3) Bi-gram and tri-gram feeatures (BT), and, (4) Beginning Of Senttence (BOS) during the training process. These features likely included lexical, syntactic, and contextual information that aided in the identification of time expressions within the text. Upon evaluation, the CRF model demonstrated a precision score of **0.79** in this task, indicating its effectiveness in accurately identifying time expressions.

To ensure the quality and accuracy of the model's predictions, the resultant labeling generated by the CRF model was subjected to manual evaluation by four annotators. Any discrepancies or errors identified during this evaluation process were addressed, and appropriate adjustments were made to the dataset as necessary. This iterative refinement process aimed to enhance the quality of the dataset and improve the performance of the CRF model in subsequent iterations.

5.2.4 TIMEX Classification

For classifying time expressions, we continued to leverage the capabilities of our Conditional Random Fields (CRF) model. Similar to the previous task, the CRF was trained on a meticulously annotated set of 600 articles and subsequently tested on the remaining 400 articles to assess its performance.

In this iteration, our CRF model incorporated an additional feature termed ISTIMEX, denoting whether or not the current word is part of TIMEX tag, along with the existing set of features. The inclusion of this feature likely provided valuable contextual information that aided in the categorization of time expressions. As a result, our CRF model achieved a precision score of 0.84 in this sub-task, indicating its efficacy in accurately categorizing the annotated time expressions.

Following the model's predictions, the labeled data underwent manual correction by four annotators across two rounds of annotation. This meticulous manual review process ensured the accuracy and reliability of the categorized time expressions within the dataset.

5.2.5 Event relation annotation

In the final phase of annotation, the resultant dataset was further enriched by manually annotating temporal links (TLINK), aspectual links (ALINK), and subordination links (SLINK). To accomplish this task, eight annotators were enlisted, and the annotation process was conducted over four rounds in batches of 125 articles each. This extensive annotation phase aimed to enhance the dataset's richness and facilitate deeper analysis of temporal relations within the text. The number of relations of each category have been compiled in Table 5.2 The final statistics of the dataset can be found in Section 5.4.

Feature	Number of occurrences
Tokens	292,517
Events	25,829
States	3,516
TIMEX	2,396
TLINK	7,289
SLINK	4,741
ALINK	433

Table 5.2: Count of Event, States, TIMEX and all types of links.

5.3 Language Specific Nuances

As our timebank represents the inaugural dataset adhering to TimeML guidelines for events and event relations in Hindi, the development of our annotation schema involved iterative refinement informed by both theoretical insights into Hindi constructions and feedback from annotators. Similar to the approach undertaken by DELTA, our objective was to devise a schema capable of yielding a high Inter-Annotator Agreement (IAA) score. Additionally, to bolster confidence in the dataset among future researchers, we incorporated an annotator confidence tag for each annotation.

During iterations of developing guidelines, the annotator confidence parameter makes it easier to justify modifications. Also, it draws attention to unclear constructions that depend on context or grammatical features that are not covered by the current rules, which could present difficulties for data processing operations down the road. The importance of annotator confidence in improving annotation procedures is highlighted by situations when it becomes especially essential, such as when subjunctives are removed from event representation.

The $\langle CONFIDENCE \rangle$ tag was proposed by Pustejovsky et al. (2008) to provide a confidence measure for every characteristic of a tag. This metric, which once had a range of 0 to 1, was meant to represent the annotator's level of confidence in attribute annotations. But this level of detail was considered too fine. It was decided that treating annotator confidence as a parameter instead of a stand-alone tag would be more appropriate in light of the attributes annotated in the Hindi TimeBank.

Our method uses a ternary annotation parameter for annotator confidence, where the annotator's level of trust in an annotation is indicated by the values HIGH, MEDIUM, and LOW. This metric is quite helpful in providing annotators with clarification on definitions, especially when it comes to classifying event and time expressions.

In addition to borrowing some ideas from existing annotation schemes, we've also made some modifications specific to Hindi. These modifications include entity-centric event descriptions, state identification, and altered event classification as a result of state categorization. This was required for a few reasons:

1. Events do not solely adhere to grammatical or linguistic principles; rather, they manifest as pragmatic occurrences depicted within text. Consequently, the linguistic representation of such phenomena tends to be contingent upon specific languages or domains.

2. In the context of Hindi, a blanket definition of events as encompassing all describable situations through language proves untenable. Such a broad classification would result in virtually any verb construction being categorized as an event, thereby rendering subsequent efforts in classification, relational establishment, and information extraction from event-annotated text futile.

We made the following language-specific modifications to TimeML annotation guidelines to adapt it to Hindi:

5.3.1 Identification of States

Events, according to TimeML, are ”situations that happen, occur, hold, or take place, as well as those predicates describing states or circumstances in which something obtains or holds true” (Pustejovsky et al., 2003a). However, Goud et al. (2019) explain the difficulties involved in explicitly annotating states and events by utilising the viewpoints of annotation rules and linguistic philosophy.

Our understanding and analysis of states and events is based on Bach's (Back, 1986) idea of states, processes, and events; which is also similar to Panini's ideas of semantic modelling of events. This is in contrast to TimeML's definition of events which seems to be motivated by the new-Davidsonian definition of an event. The need for a different notion of states can be understood using the following example:

1. *pa:iso ki* ***kamI*** *se Ram ke parivaar ko bohat takalipha honi lagi*

2. *pai:so ki* ***kamI hone*** *se Ram ke parivaar ko bohat takalipha honi lagi*

As you can see, although these sentences convey similar meanings, the syntactic structure of the subordinate verb clause varies significantly due to the presence of the verbal auxiliary ”hone,” which explicitly conveys a telic and a durative situation. TimeML guidelines stipulate that generics and verbal clauses featuring generic arguments should not be annotated as events. However, the auxiliary ”hone” is employed with generics to create semantically equivalent sentences. Consequently, according to TimeML annotation standards, annotators would label ”kamI” as an event, but ”kamI hone” would be marked as such, despite both being used similarly within the sentence. Such constructions if skipped, cause confusion for annotators. We employ the Paninian framework to resolve this. In the example shared above, we decide to uniformly mark both *kami* and *kami hone* as DESC (descriptive) states. The reasoning behind this decision can be understood using Bach's classification of habitual verbal predicates and other semantically equivalent forms. We can summarize it as follows: Verbal auxiliaries play a dual role in providing both syntactic and semantic information regarding the verbal predicate, which is essential. Thus, a verbal predicate can be classified as either a state or an event when compared to Bach's concept of eventualities.

5.3.2 Classification Mechanism

Since both states and events are undergoing annotation as distinct concepts, the classification outlined in TimeML (Pustejovsky et al., 2003a) cannot be directly applied. Consequently, the STATE and I-STATE event categories have been excluded from consideration. Our examination has revealed disparities between our understanding of states and the TimeML representation thereof. In TimeML, I-STATE pertains to states referencing alternate or hypothetical scenarios, which Hindi typically conveys through subjunctive constructions devoid of explicit participants. Therefore, by linguistic definition, I-STATE instances are not classified as states but rather identified as OCC events.

To address this discrepancy, we propose a novel classification schema for states, distinguishing between DESC and DECL categories. I've described the two classes in detail in Section 5.1.2.

5.3.3 Marking Time Expressions

Time Expressions in all TimeML datasets are annotated using the TIMEX3 notation. TIMEX3 analysis in Hindi has been explored in prior works (Ramrakhiyani and Majumder, 2013; Ramrakhiyani and Majumder, 2015) to assess the identification and classification of time expressions. A departure from conventional practice involves annotating fragmented time expressions solely with tokens denoting local time, consolidating them under a single TIMEX id. Relative time expressions like ”cara sala” (four years) receive TIMEX annotation only if their duration can be reasonably estimated. Additionally, the annotation of time expressions incorporates dependency and semantic role information, aspects not accounted for in TIMEX3.

5.4 Dataset Statistics

In this section, we present an analysis of the distribution of event and state categories within the dataset. Table 5.3 provides a breakdown of the distribution of events, revealing that the occurrence type (OCC) is the most prevalent, constituting 87.52% of all events. The aspectual type (ASP) comprises 1.62%, the intensional action (IAC) accounts for 3.03%, perception events (PER) represent 1.62%, and reporting events (REP) make up 6.19% of the total events. Table 5.3 also reveals that the majority of time expressions belong to the DATE class.

The prevalence of the occurrence type is attributed to its broad classification criteria and the absence of strict syntactic and semantic constraints. Events were categorized as occurrences if they did not fit into any other category.

Similarly, we analyze the distribution of states, revealing that 53.04% are descriptive (DESC) and 46.96% are declarative (DECL) in nature.

	Category	Total
Event	OCC	22,606
	REP	1,599
	IAC	783
	ASP	421
	PER	420
	Total	**25,829**
State	DESC	1,865
	DECL	1,651
	Total	**3,516**
TIMEX	DATE	1,390
	DUR	545
	TIME	433
	SET	28
	Total	**2,396**

Table 5.3: Category-wise distribution of Events, States and Time Expressions in Hindi Timebank

In the final corpus, we break down annotator confidence by category in Table 5.4. Notably, confidence levels vary most for events that blur the line between events and states. Verbal predicates with only tense auxiliaries or those in light verb constructions tend to elicit lower confidence, particularly if they pertain to fragmented events. Descriptive states also evoke lower confidence, suggesting ambiguity or difficulty in classification. However, TIMEX classification generally yields higher confidence scores. There are some instances of lower and medium confidence in TLINKS and ALINKS classifications. Among SLINKS, links involving the subordination of OCC-OCC are particularly ambiguous, leading to lower confidence among annotators.

Category	High	Medium	Low
Event Categories	92.94%	5.86%	1.90%
State Categories	91.07%	5.52%	3.41%
TIMEX Categories	95.69%	4.31%	0.00%
TLINK	90.86%	4.25%	4.89%
ALINK	93.35%	4.60%	2.05%
SLINK	89.77%	5.71%	4.52%

Table 5.4: Category-wise breakdown of Annotator Confidence Scores in Hindi Timebank

Chapter 6

Conclusion

In conclusion, this thesis represents a significant step forward in the field of temporal relation extraction, aiming to enhance the robustness and generalizability of existing models. By addressing challenges such as low inter-annotator agreement scores and the prevalence of vague relations, this work contributes to a deeper comprehension of how machines can interpret and reason about temporal information within natural language text. The development of DELTA 2.0, an annotated dataset capturing global temporal relations, along with the creation of an annotation schema and tool to mitigate issues related to vague relations, underscores the commitment to advancing the field. Additionally, the release of Adapted-TIMERS as an open-source model further facilitates research and collaboration in this domain. Finally, the establishment of the Hindi TimeBank as an ISO-TimeML Annotated Reference Corpus aims to foster exploration and innovation in temporal relation extraction for Indic languages and beyond.

6.1 Future Work

This thesis has made significant contributions to the field of Temporal Relation Extraction (TRE) by addressing key challenges and proposing novel methods. Building upon this strong foundation, several promising avenues exist for future research:

6.1.1 Leveraging Discourse Information for Improved Reasoning

This thesis highlights the importance of discourse-level information for accurate TRE. Future work can explore more sophisticated methods for incorporating discourse features, such as:

1. Coreference Resolution: Identifying and linking coreferent expressions within the document can provide valuable contextual information for understanding temporal relationships.
2. Event Argument Role Labeling: Extracting the semantic roles of event arguments (e.g., agent, patient) can offer further insights into the temporal ordering between events.

3. Discourse Coherence Relations: Analyzing the discourse coherence relations between sentences (e.g., elaboration, contrast) may provide clues about the temporal flow of information.

Research on models that integrate these discourse features with existing TRE methods holds promise for achieving more robust and accurate performance.

6.1.2 Handling Vague Temporal Expressions

Vague temporal expressions remain a significant challenge for TRE models. Promising future directions include:

1. Fuzzy Logic Integration: Incorporating fuzzy logic techniques can allow models to handle uncertainty associated with vague expressions like ”soon” or ”later.”

2. Temporal Reasoning with Contextual Cues: Developing models that reason about temporal expressions by considering surrounding words and phrases can improve the interpretation of vagueness.

3. Utilizing Event Type Information: The type of event (e.g., meeting, travel) may provide context clues that help disambiguate vague temporal expressions. These approaches hold promise for improving the ability of TRE models to cope with the inherent ambiguity of natural language when describing time.

6.1.3 Cross-lingual Transfer for Multilingual TRE

The Hindi TimeBank corpus developed in this thesis serves as a valuable resource for exploring multilingual TRE. Future work can investigate:

1. Domain Adaptation Techniques: Developing techniques for adapting models trained on English to new domains within Hindi or other languages can enhance cross-lingual performance.

2. Multilingual Pre-trained Language Models: Leveraging pre-trained language models that capture temporal information across multiple languages can facilitate effective cross-lingual TRE.

3. Exploiting Parallel Corpora: Utilizing parallel corpora where documents are available in both source and target languages can provide valuable training data for cross-lingual TRE models.

By exploring these directions, research can advance the state-of-the-art for TRE tasks in languages beyond English.

To sum up, this thesis has laid a strong foundation for future advancements in TRE. By focusing on discourse information, handling vagueness, and exploring cross-lingual approaches, researchers can continue to push the boundaries of what is possible in this crucial field. References to existing promising work, such as coreference resolution and fuzzy logic techniques, provide concrete starting points for future research endeavors. As the field progresses, TRE models will become increasingly sophisticated in their ability to understand and reason about temporal relations within natural language text.

Related Publications

1. Pranav Goel, Suhan Prabhu, Alok Debnath, **Priyank Modi**, and Manish Shrivastava. 2020. Hindi TimeBank: An ISO-TimeML Annotated Reference Corpus. *(Accepted)*

2. **Priyank Modi**, Ujwal Narayan and Manish Shrivastava. DELTA: A dataset for discourse level event timeline generation. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP). Submitted to EMNLP 2022 *(Borderline Reject - 3/5)*

3. **Priyank Modi**, Yatin Nandwani, Maneesh Singh. 2024. [Re] TIMERS: Document-level Temporal Relation Extraction. Under Review in Rescience Journal *(In Review)*

4. **Priyank Modi**, Manish Shrivastava, and Ujwal Narayan. DELTA: A dataset for discourse level event timeline generation. Submitted to CIKM: Conference on Information and Knowledge Management. 2024. *(In Review)*

Bibliography

2001. *Proceedings of the ACL 2001 Workshop on Temporal and Spatial Information Processing.*

2006. *Pustejovsky, James, et al. TimeBank 1.2 LDC2006T08. Web Download. Philadelphia: Linguistic Data Consortium, 2006.*

Miguel Ballesteros, Rishita Anubhai, Shuai Wang, Nima Pourdamghani, Yogarshi Vyas, Jie Ma, Parminder Bhatia, Kathleen R. McKeown, and Yaser Al-Onaizan. 2020. Severing the edge between before and after: Neural architectures for temporal ordering of events. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5412–5417. Association for Computational Linguistics.

André Bittar, Pascal Amsili, Pascal Denis, and Laurence Danlos. 2011. French TimeBank: An ISO-TimeML annotated reference corpus. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 130–134, Portland, Oregon, USA. Association for Computational Linguistics.

Philip Bramsen, Pawan Deshpande, Yoong Keok Lee, and Regina Barzilay. 2006. Inducing temporal graphs. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 189–198, Sydney, Australia. Association for Computational Linguistics.

Tommaso Caselli, Valentina Bartalesi Lenzi, Rachele Sprugnoli, Emanuele Pianta, and Irina Prodanof. 2011. Annotating events, temporal expressions and relations in Italian: the it-timeml experience for the ita-TimeBank. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 143–151, Portland, Oregon, USA. Association for Computational Linguistics.

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

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

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*.

Muriel Ekdahl and Joseph E Grimes. 1964. Terena verb inflection. *International Journal of American Linguistics*, 30(3):261–268.

JENNIFER R. ELLIOTT. 2000. Realis and irrealis: Forms and concepts of the grammaticalisation of reality. 4(1):55–90.

Harry Feldman et al. 1986. *A grammar of Awtuw*. Dept. of Linguistics, Research School of Pacific Studies, The Australian ....

Jaipal Singh Goud, Pranav Goel, Alok Debnath, Suhan Prabhu, and Manish Shrivastava. 2019. A semantico-syntactic approach to event-mention detection and extraction in hindi. In *Workshop on interoperable semantic annotation (ISA-15)*, volume 63.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019. Joint event and temporal relation extraction with shared representations and structured prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 434–444. Association for Computational Linguistics.

Rujun Han, Xiang Ren, and Nanyun Peng. 2020. DEER: A data efficient language model for event temporal reasoning. *CoRR*, abs/2012.15283.

Thorsten Joachims, Thomas Hofmann, Yisong Yue, and Chun-Nam John Yu. 2009. Predicting structured objects with support vector machines. *Commun. ACM*, 52(11):97–104.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Kenton Lee, Yoav Artzi, Yejin Choi, and Luke Zettlemoyer. 2015. Event detection and factuality assessment with non-expert supervision. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1643–1648.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in*

Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, pages 188–197. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Diego Marcheggiani, Michael Roth, Ivan Titov, and Benjamin Van Durme. 2017. Semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, Copenhagen, Denmark. Association for Computational Linguistics.

Puneet Mathur, Rajiv Jain, Franck Dernoncourt, Vlad Morariu, Quan Hung Tran, and Dinesh Manocha. 2021. TIMERS: Document-level temporal relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 524–533, Online. Association for Computational Linguistics.

Teruko Mitamura, Yukari Yamakawa, Susan Holm, Zhiyi Song, Ann Bies, Seth Kulick, and Stephanie Strassel. 2015. Event nugget annotation: Processes and issues. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 66–76.

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.

Qiang Ning, Hao Wu, Haoruo Peng, and Dan Roth. 2018a. Improving temporal relation extraction with a globally acquired statistical resource. *arXiv preprint arXiv:1804.06020*.

Qiang Ning, Hao Wu, and Dan Roth. 2018b. 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. 2018c. CogCompTime: A tool for understanding time in natural language. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 72–77, Brussels, Belgium. Association for Computational Linguistics.

Keiron O'shea and Ryan Nash. 2015. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Sauparna Palchowdhury, Prasenjit Majumder, Dipasree Pal, Ayan Bandyopadhyay, and Mandar Mitra. 2011. Overview of fire 2011. In *Fire*.

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, Robert Ingria, Roser Sauri, José M Castaño, Jessica Littman, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Inderjeet Mani. 2005. The specification language timeml.

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).

Roser Saurí and James Pustejovsky. 2009. Factbank: a corpus annotated with event factuality. *Language resources and evaluation*, 43(3):227–268.

Alex Sherstinsky. 2020. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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.

Marc Verhagen, Roser Saurí, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, Uppsala, Sweden. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020. Joint constrained learning for event-event relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 696–706. Association for Computational Linguistics.

Yadollah Yaghoobzadeh, Gholamreza Ghassem-Sani, Seyedabolghasem Mirroshandel, and Mahbaneh Eshaghzadeh Torbati. 2012. Iso-timeml event extraction in persian text. pages 2931–2944.

Xinyu Zhao, Shih-Ting Lin, and Greg Durrett. 2020a. Effective distant supervision for temporal relation extraction. *CoRR*, abs/2010.12755.

Xinyu Zhao, Shih-ting Lin, and Greg Durrett. 2020b. Effective distant supervision for temporal relation extraction. *arXiv preprint arXiv:2010.12755*.