

Temporal Relation Extraction in Clinical Texts: A Span-based Graph Transformer Approach

Rochana Chaturvedi Peyman Baghershabi Sourav Medya Barbara Di Eugenio

University of Illinois Chicago

Chicago, IL, U.S.A.

{rchatu2, pbaghe2, medya, bdieugen}@uic.edu

Abstract

Temporal information extraction from unstructured text is essential for contextualizing events and deriving actionable insights, particularly in the medical domain. We address the task of extracting clinical events and their temporal relations using the well-studied I2B2 2012 Temporal Relations Challenge corpus. This task is inherently challenging due to complex clinical language, long documents, and sparse annotations. We introduce GRAPH-TREX, a novel method integrating span-based entity-relation extraction, clinical large pre-trained language models (LPLMs), and Heterogeneous Graph Transformers (HGT) to capture local and global dependencies. Our HGT component facilitates information propagation across the document through innovative global landmarks that bridge distant entities and improves the state-of-the-art with 5.5% improvement in the temporal F_1 score over the previous best and up to 8.9% improvement on long-range relations, which presents a formidable challenge. We further demonstrate generalizability by establishing a strong baseline on the E3C corpus. Not only does this work advance temporal information extraction, but also lays the groundwork for improved diagnostic and prognostic models through enhanced temporal reasoning.

1 Introduction

Electronic Health Records (EHRs) contain a wealth of information in both structured data and unstructured free-text notes written by healthcare providers. These notes are the preferred information source of physicians as they contain valuable insights often missing from the structured data (Hersh et al., 2013; Capurro et al., 2014), including temporal progression of symptoms, diagnoses, and treatments. Modeling such temporal relations is important for understanding the underlying disease pathology. For example, a drug given *after* a symptom appears may indicate a treatment, while a

symptom appearing *after* a drug may indicate an adverse effect. Precise extraction of event chronology from text can aid precision medicine by improving temporal reasoning for downstream applications (Chen et al., 2021; Vashishtha et al., 2020).

One such application is facilitating *opportunistic screening* of chronic health conditions such as Diabetes Type 2 (T2D) (Pickhardt et al., 2021; Scheetz et al., 2021). Many underprivileged patients forgo regular check-ups but interact with the healthcare system for other reasons (Zhang et al., 2015; Dhippayom et al., 2013). Opportunistic screening—screening patients for certain conditions whenever they come in contact with the healthcare system, possibly for unrelated reasons—is expensive at scale. Leveraging patient histories from the clinical notes can enable this by identifying early risk factors, thereby enhancing healthcare equity and reducing clinicians’ manual review burden. This requires accurate temporal relation extraction (TREX) to determine when risk factors emerge in a patient’s history. Beyond risk factors, TREX also plays a crucial role in understanding patterns of drug use and misuse, as well as assessing the long-term benefits and potential side effects of medications. For instance, the shared task by Yao et al. (2024) on extracting cancer treatment trajectories highlights the growing significance of this vital research area. Modeling events and their temporal connections has shown promise for event forecasting in other fields (Li et al., 2021), but no comparable work has been applied to the clinical domain. While some patient representation models consider coarse temporal chronology across patient visits (Ghassemi et al., 2015; Lee et al., 2020; Chaturvedi et al., 2023), fine-grained temporal information in the clinical notes remains underutilized (Tvardik et al., 2018).

The clinical TREX task faces at least three challenges. First, clinical text requires domain-specific models due to its specialized vocabulary (e.g.,

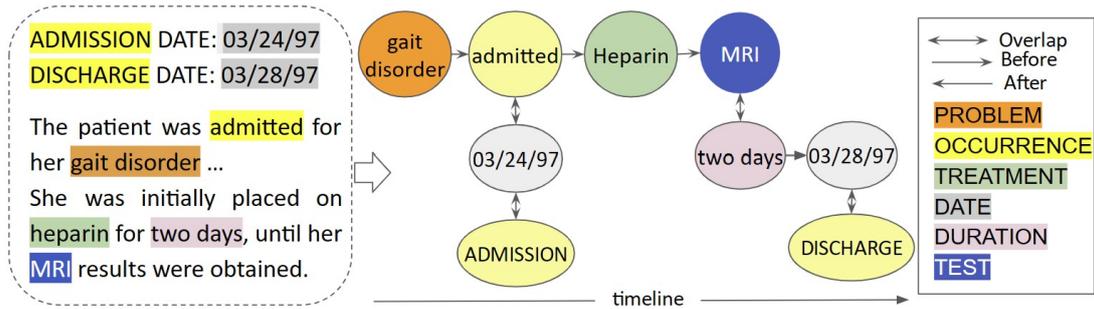


Figure 1: Temporal graph of clinical events and time expressions from a clinical note with color-coded spans as nodes and edges indicating their temporal relations.

‘b.i.d.’ meaning twice a day for dosage), unique temporal expressions (e.g., ‘three days postop’), and ambiguity in abbreviations across providers (e.g. ‘DM2’ can mean Diabetes Mellitus Type 2 or Myotonic Dystrophy Type 2), requiring contextual interpretation. However, publicly available, moderate-sized, free or affordable corpora of clinical notes annotated with temporal relations are scarce due to costly expert annotation and strict privacy regulations. Several works use the I2B2 2012 corpus (Sun et al., 2013a,b), as we do here. Another corpus, THYME (Bethard et al., 2017; Styler IV et al., 2014), requires a membership fee similar to the Linguistic Data Consortium (LDC). Fortunately, a small corpus annotated with THYME guidelines, E3C (Magnini et al., 2022), was recently released, and we report results on it as well.

Second, TREX is often tackled using a sequential (pipeline) approach that first extracts entities and then classifies the entity pairs into a relation type (Xu et al., 2013). Unlike other relation extraction tasks, temporal relations are constrained by temporal logic and often implied through linguistic cues (verb tenses, temporal prepositions) and reasoning over the extracted relation pairs. Conventional sequential approaches struggle to capture such interactions between entities and temporal relations, often leading to error propagation.

Third, clinical notes generally comprise long documents with extensive patient history, findings, treatment plans, etc. However, the recent transformer-based models that perform exceptionally well in NLP tasks have context-length limitations. Due to this, recent works in clinical TREX target entity pairs across small context lengths, such as only event pairs across three consecutive sentences (Han et al., 2020) to avoid the quadratic complexity of document-level context, overlooking both the long-distance pairs and interactions

with the time expressions. Consider the example in Figure 1, containing a small excerpt of a real clinical note and corresponding temporal graph. It is easy to infer that gait disorder < (*before*) admission < heparin < MRI, by observing the local context. Given these patterns, one can also infer the long-range relation that gait disorder < MRI. Inferring such long-range relations is essential to obtain more accurate timelines; however, it is known to be a formidable challenge (Qin et al., 2023a).

Contributions. To address these challenges, we introduce GRAPHTREX (**Graph-based Temporal Relation Extraction**), an end-to-end framework integrating text- and graph-based methods, enabling a more structured and context-aware extraction. GRAPHTREX constructs a document-level Heterogeneous Graph (HG) from span-based model predictions using a clinical Pre-trained Language Model (PLM), then applies Heterogeneous Graph Transformers (HGT) for local and global information propagation. By leveraging tailored node types and heterogeneous information aggregation, GRAPHTREX achieves state-of-the-art performance on the I2B2 2012 corpus and establishes a competitive novel baseline on the E3C corpus. Importantly, our ablation experiments demonstrate that GRAPHTREX excels in inferring long-distance relations, a critical challenge in clinical TREX.

2 Related Work

Early clinical TREX approaches use a hybrid of rule-based and machine-learning-based systems that require extensive feature engineering and additional annotations (Xu et al., 2013; Tang et al., 2013; Sohn et al., 2013). Transformer-based PLMs such as BERT (Devlin et al., 2019) improve both entity extraction (Alsentzer et al., 2019; Si et al., 2019) and relation classification (Zhou et al., 2021) to classify known gold-standard entity pairs, ex-

cluding unrelated pairs. However, they have not been applied to the full end-to-end extraction task due to document-level complexity, and remain limited to nearby event-event pairs (Han et al., 2020). Since this task remains essential for real clinical use cases, we refocus on the complete document-level, end-to-end TREX using sliding windows and graph-based modeling to incorporate long contexts.

Contemporary clinical TREX methods have not shown improvement over the older baselines. Most of the works adopt traditional token-based models for entity extraction (Liu et al., 2017), which are prone to cascading label misclassifications (Yu et al., 2022). Secondly, the pipeline approaches for end-to-end relation extraction struggle to capture the complex interactions between entities and their relationships. In the general domain, joint span-based approaches (Dixit and Al-Onaizan, 2019; Lai et al., 2021; Eberts and Ulges, 2020; Yan et al., 2022) have been shown to address these limitations by modeling entity and relation extraction together. They also mitigate the cascading error issue of token-based systems by enumerating all possible contiguous spans and then directly classifying them. Another joint framework REBEL (Huguet Cabot and Navigli, 2021) that generates entity-relation triplets autoregressively has shown strong performance in document-level relation extraction. However, it underperforms in clinical TREX (Saiz and Altuna, 2023), likely due to the unique challenges posed by temporal dependencies. We integrate a span-based approach with structured graph reasoning to address these challenges.¹

3 Problem Statement & Data

3.1 Problem Statement

Given a clinical note, we extract a temporal graph \mathcal{G} , jointly identifying the nodes and edges. The nodes correspond to entities, which comprise either clinical events such as symptoms or medications; or time expressions (timex) such as dates, frequencies, or durations. Edges represent temporal relations. As shown in Figure 1, a directed edge from e_1 to e_2 represents the temporal relation e_1 *Before* ($<$) e_2 or e_2 *After* ($>$) e_1 while an undirected edge represents e_1 *Overlaps* ($=$) e_2 .

3.2 Data

We primarily focus on the I2B2 2012 Temporal Relations Challenge corpus (Sun et al., 2013a,b)

¹See Appendix S1 for additional related works.

in this study, which features a substantial number of cross-sentence annotations, unlike other clinical TREX corpora. The data comprises 310 discharge summaries in English where entities include:

Clinical Events: Events include six types—, namely TREATMENT, TEST, PROBLEM, CLINICAL DEPARTMENT, EVIDENTIAL, and OCCURRENCE. While TREATMENT, TEST, and PROBLEM are self-explanatory; EVIDENTIAL denotes information source (e.g. ‘tested’, ‘complained’); OCCURRENCE refers to activities such as transfers between departments, admission, discharge, or follow-up. CLINICAL DEPARTMENT anchors key services in the patient’s timeline, often reflecting changes in severity (e.g., ICU admission indicates higher severity).

Time Expression (TimEx): Includes DATE, TIME, DURATION, and FREQUENCY.

SECTIME: Time anchors marking section creation times, like ADMISSION (clinical history) or DISCHARGE (hospital course) dates.

Temporal relations or Tlinks: Include *Before* ($<$), *After* ($>$), and *Overlap* ($=$). TLinks can exist between event-event (EE), event-timex or event-sectime (ET), and timex-timex (TT).

Train-test Split. The data is divided into 190 training files and 120 test files. Following (Zhou et al., 2021), we sample 9 files from the training set as the development set. On average, a document contains 86.6 events, 12.4 timexes, and 176 TLinks. There’s an average of 576 tokens per document, 110 entities, and 197 annotated relations.

We also conduct initial experiments on the English subset of the E3C corpus (Magnini et al., 2022), a collection of clinical cases. The dataset includes 48 test documents and 36 training documents, with 7 (20%) reserved for development.

Appendix S2 contains additional data details.

4 Our Method: GRAPHTREX

The clinical texts are long documents with numerous entities and relations. The main motivation for using graphs is that these long documents contain implicit temporal dependencies that extend beyond local neighborhoods, making them difficult to capture with span-based approaches alone. To discover the global patterns, we create heterogeneous graphs from whole documents, with entities as nodes and their temporal relations as edges. We include special landmark nodes to incorporate local and global context. Our model GRAPHTREX works in three

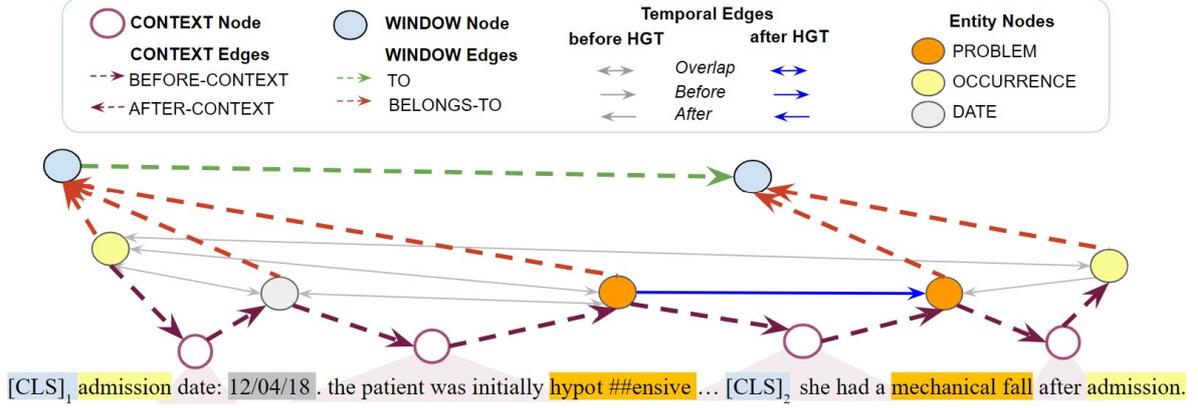


Figure 2: Heterogeneous Graph Construction. The gray edges represent intermediate high-confidence temporal relations inferred before HGT integration. The blue arrow represents additional relation inferred by the HGT component through local neighborhood aggregation via the entity and CONTEXT nodes and global neighborhood aggregation via the WINDOW nodes.

steps—(i) obtaining domain-aware representations, (ii) constructing a heterogeneous graph, and (iii) obtaining final predictions with GNN.

4.1 Domain-aware Representations (SPANTREX)

We adopt a span-based approach to obtain domain-aware features, where a span is a contiguous sequence of tokens to be classified as an entity. We enumerate all spans up to length $k = 7$, covering the 97th percentile of entity lengths in the training data. Span representations are derived using BioMedBERT (Gu et al., 2021), pre-trained from scratch on biomedical articles to capture domain-specific vocabulary. For example, BioMedBERT recognizes ‘creatinine’ as a single token unlike general-domain BERT, which splits it into less meaningful subwords [‘cr’, ‘##ea’, ‘##tin’, ‘##ine’]. The representation \mathbf{e}_{sp} of a span sp is obtained by concatenating BioMedBERT embeddings ρ for the span’s start token ($\text{start}[sp]$), end token ($\text{end}[sp]$), and span-width embedding ($\omega(sp)$) encoding the number of tokens in a span ($|sp|$).

$$\mathbf{e}_{sp} = FFNN(\rho_{\text{start}[sp]}; \rho_{\text{end}[sp]}; \omega(sp)) \quad (1)$$

A feed-forward network with two layers and ReLU activation is used as an entity decoder.

$$\hat{y}_i^n = \operatorname{argmax}_{\mathcal{A}_{sp}}(g(\mathbf{e}_{sp_i})) \quad (2)$$

$$g(\mathbf{e}_{sp_i}) = \operatorname{softmax}(FFNN(\mathbf{e}_{sp_i})) \quad (3)$$

where \mathcal{A}_{sp} is the set of entity/span types, including the NOT-ENTITY class for the remaining spans.

For relation classification, we create pair-wise representations using entities not classified as NOT-

ENTITY. We concatenate each entity span representation with the element-wise product of the two spans to capture the interactions between the spans. We also include the max-pooled representation of all the token embeddings between two entity spans. This design choice is inspired by several previous studies that observe significant performance improvement in relation extraction tasks by providing additional context between two entities (Cheng and Weiss, 2023; Li et al., 2023a)—i.e. \mathbf{e}_{ctx} denoted the context between tokens i and j . Additionally, we also provide the entity types as inferred by the entity decoder to the model for modeling domain-specific distributional constraints (Han et al., 2020; Cheng and Weiss, 2023):

$$\begin{aligned} \epsilon_{i,j} &= [\mathbf{e}_{sp_i}; \mathbf{e}_{sp_j}; \mathbf{e}_{sp_i} \odot \mathbf{e}_{sp_j}; \mathbf{e}_{\hat{y}_i^n}; \mathbf{e}_{\hat{y}_j^n}; \mathbf{e}_{\text{ctx}(i,j)}] \\ \mathbf{e}_{\text{ctx}(i,j)} &= \operatorname{maxpool}(\{\rho_k \mid \text{end}[sp_i] < k < \text{start}[sp_j]; i < j\}) \end{aligned} \quad (4)$$

A relation decoder similar to the entity decoder is used to classify the pair representation of i^{th} and j^{th} spans into $\mathbf{Y}_{\text{rtype}} \in \{\text{Before}, \text{After}, \text{Overlap}, \text{NO-RELATION}\}$.

$$\hat{y}_{ij}^r = \operatorname{argmax}_{\mathcal{R}_{sp}}(f(\epsilon_{i,j})) \quad (5)$$

$$f(\epsilon_{i,j}) = \operatorname{softmax}(FFNN(\epsilon_{i,j})) \quad (6)$$

where \mathcal{R}_{sp} is the set of relation types.

For our task of joint end-to-end relation extraction, we use the combined training loss $\mathcal{L} = \mathcal{L}_n + \mathcal{L}_r$. Where \mathcal{L}_n is the cross-entropy loss for entity and \mathcal{L}_r for relation extraction.

Sliding Windows. Clinical notes are long, and BioMedBERT is limited to 512 tokens. Previous

works often split the document into 1–3 sentence chunks, processing each separately, which results in loss of global context and potential inconsistencies. To overcome this, we use a sliding window approach to retain sufficient local context for each token’s embedding (Beltagy et al., 2020). The sliding window mechanism processes a sequence by moving a fixed-size (n tokens) window with a stride of $\lfloor \frac{n-2}{2} \rfloor$. The window masks ensure that despite overlapping windows to provide sufficient context to all tokens, the final representation of each token comes from a unique window (See Appendix S3 for details). These local representations are then used for TREX.

We now describe our model GRAPHTREX that uses the predicted entities and their most confident Tlinks and refines the extracted temporal graph using graph neural networks (GNN).

4.2 Document-Level Heterogeneous Graph

A Heterogeneous Graph (HG) models heterogeneities in both the node and the edge types (Shi et al., 2017). It is defined as a tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$, where $\mathcal{V} = \{v_i\}_{i=1}^{N_v}$ and $\mathcal{E} = \{e_j\}_{j=1}^{N_e}$ are the sets of nodes and edges. The node types and edge types are denoted by \mathcal{A} and \mathcal{R} , with a unique mapping for each node $\phi(v) : \mathcal{V} \rightarrow \mathcal{A}$ and edge $\psi(e) : \mathcal{E} \rightarrow \mathcal{R}$. Here N_v and N_e represent the total number of nodes and edges. We can model the graph as a set of triples $(\phi(s), \psi(e), \phi(t))$, or “meta relations”, for every source node s connected to target node t by edge e . To construct an HG from the documents (illustrated in Figure 2), we define two categories of node types, namely, *Token-level* nodes (Entity/CONTEXT) and *Document-level* nodes (WINDOW) to incorporate local and global context, allowing information exchange between both the neighboring entities and across the windows. We describe these in the following paragraphs and further elaborate in Appendix S4.

Token-level Nodes. Node types in this group capture the contextual understanding of the spans and their local inter-dependencies. These include *Span-based* and *Context-based* nodes.

Entity Nodes. We represent every span that is not predicted as NOT-ENTITY by g as a node in this subgroup. The node type for these is the predicted entity type. Thus, we obtain $|\mathcal{A}_{sp}| - 1$ node types initialized with their representation e_{sp} . We connect two such nodes if f predicts their corresponding temporal relation other than NO-RELATION,

with the prediction probability above a pre-defined threshold τ . This **confidence thresholding** ensures that information is propagated only via the most certain relation predictions of f . This is particularly important in the early stages of training, as all components are trained simultaneously without certainty filtering, a sub-optimal model might propagate the noise across the graph.

Context-based Nodes. In natural language, the meaning of a word or phrase is closely tied to its local context (Radford et al., 2018; Devlin et al., 2019; Mikolov et al., 2013). We also observe this in our experiments with SPANTREX, where incorporating additional context between spans brings a high-performance gain in relation extraction (see Figure 4). To capture this in the HG, we introduce *context nodes* that store pooled context between a span pair. Directed edges connect the spans to the context node in the lexical order. Introducing context nodes for all possible span pairs results in a dense graph where even far-apart spans are reachable within 1- or 2-hops, leading to the common over-smoothing issue after information aggregation. To mitigate this, we limit context nodes to span pairs within a distance of δ tokens.

Document-level Nodes. To detect the long-range dependencies, we introduce window nodes which act as global landmarks in the document. Given a common window length L_w for all documents, there can be $\lceil \frac{L}{L_w} \rceil$ window nodes in a document of length L while the node type WINDOW is assigned to all of them. Entity nodes within a window connect to their window node via distinct edge types based on the entity types, and window nodes are connected sequentially based on their lexical order. Window nodes are of significant importance since they help aggregate information from far neighborhoods and help in finding transitive relations.

4.3 Final Predictions by GRAPHTREX

Once an HG is constructed, we model it with the heterogeneous graph transformer (HGT) (Hu et al., 2020). Unlike traditional GNN typically applied to homogeneous graphs (Veličković et al., 2018), HGT considers node- and edge-type distributions. The HGT module takes a document graph \mathcal{G} constructed as discussed in Section 4.2 and initializes entity nodes with span embeddings (e_{sp}). To produce structure-enhanced representations, the HGT module applies a message-passing process and enriches span embeddings with the aggregated in-

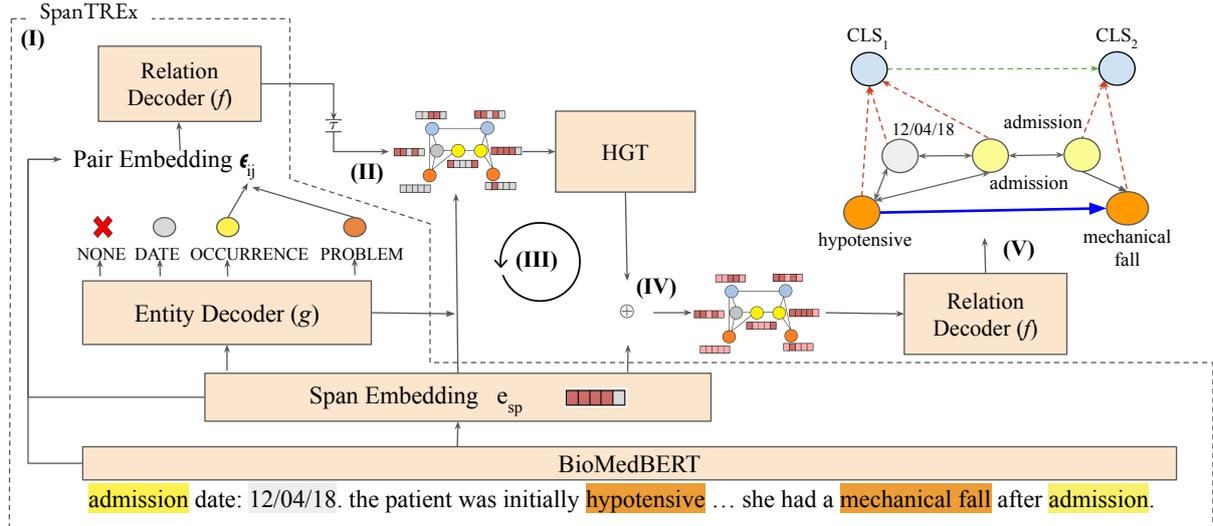


Figure 3: GRAPHTREX Architecture. (I) The dashed boundary represents SPANTREX that produces the initial temporal graph. (II) A confidence threshold (τ) filters uncertain edges (TLinks). (III) The edges are iteratively refined with the HGT component. (IV) The final node representations from the graph are combined with weighted residual of original span embeddings. (V) The relation decoder predicts the final temporal relations based on the HGT-enhanced representation. This way GRAPHTREX infers additional local and global relations (blue arrow).

formation from their neighborhoods in the graph to extract global relation patterns effectively (the HGT module is described in Appendix S4.1).

Figure 3 shows the complete GRAPHTREX architecture. After obtaining HGT representations for entities, we add them to the weighted residual of the original span embeddings. The new representations are then fed to the relation decoder. Here the nodes with CLS labels refer to the Window nodes (we do not show the context nodes for clarity). The example highlights that while SPANTREX can infer the *Overlap* ($=$) relation between co-referring mentions of “admission”, HGT integration in GRAPHTREX allows us to go further. For example, by aggregating information that ‘hypotensive’ = ‘admission’ and ‘mechanical fall’ $>$ ‘admission’, HGT helps detect the long-range relation ‘hypotensive’ $<$ ‘mechanical fall’ as shown with the blue arrow.

5 Experiments

5.1 Reproducibility

The implementation details are in Appendix S5, and our code is available at <https://github.com/pbagherashahi/GraphTREX>.

5.2 Evaluation Setup

The event extraction and timex extraction tasks in the I2B2 2012 challenge are evaluated using span F_1 (EI for entity identification) rewarding partial span match, and entity type accuracy (EC for en-

tity classification). For the end-to-end relation extraction (RE), the TLink F_1 is considered as the primary metric. Following the I2B2 2012 shared task, we exclude the NO-RELATION pairs for final evaluation. An important consideration while evaluating Temporal Relation Extraction (TREX) methods is that the same temporal order of events can be defined in multiple ways. For example, ($A < B$, $B = C$) can also be specified as ($A < C$, $B = C$) due to transitivity, and ($A < B$) is the same as ($B > A$) due to invertibility. Therefore, the temporal awareness metrics (Uzzaman and Allen, 2011) are adopted for evaluation:

$$P = \frac{|G_{sys}^- \cap G_{gold}^+|}{G_{sys}^-}, \quad R = \frac{|G_{gold}^- \cap G_{sys}^+|}{G_{gold}^-}, \quad F_1 = 2 \frac{PR}{P+R}$$

where G^+ is the closure of graph G that makes implicit temporal relations explicit by reasoning on a set of relations using transitivity and invertibility rules. G^- is the reduced graph after removing redundant relations inferable from other relations of G . The intersections in the formulae refer to the TLinks of the corresponding graphs. For E3C corpus, we use micro-F1 scores for both entity and relations following the THYME standard.

Baselines. For the I2B2 corpus, we evaluate GRAPHTREX and SPANTREX against hybrid rule and ML-based approaches (Xu et al., 2013; Tang et al., 2013), a recent generative framework (Saiz and Altuna, 2023), popular general-domain model SPERT (Eberts and Ulges, 2020), and multi-head

Model	EVENT		TimEx		TLink		
	EI	EC	EI	EC	RE		
	F_1	Acc	F_1	Acc	P	R	F_1
Rule+ML (Xu et al., 2013)	91.66	85.74	91.44	89.29	67.10	60.01	63.36
Rule+ML (Tang et al., 2013)	90.13	83.60	86.59	85.00	70.06	56.88	62.78
REBEL+BART (Saiz and Altuna, 2023)	78.00	72.00	77.00	65.00	65.00	52.00	58.00
Multi-head Attention (Miller et al., 2023)	89.42	80.74	89.01	77.42	86.12	29.76	44.24
SPERT (Eberts and Ulges, 2020)	89.44	81.1	90.29	82.64	77.58	50.36	61.08
SPANTREX (ours)	89.49	81.35	90.13	81.32	81.33	56.44	66.63
GRAPHTREX (ours)	89.55	80.99	90.06	81.21	78.21	61.42	68.81

Table 1: I2B2 2012 end-to-end entity-relation extraction task results using TLink F_1 as the primary metric.

attention-based method having state-of-the-art on the THYME corpus Miller et al. (2023).² We also experiment with a pipeline approach—extracting entities using our state-of-the-art BioMedBERT-UMLS model (Appendix S6.1) and classifying entity pairs using the method from Zhong and Chen (2021), which attains state-of-the-art in I2B2 2012 relation classification (Cheng and Weiss, 2023). We extend this to include the majority NO-RELATION class for pairs without clear temporal links. Similar to earlier works, we are unable to run this model for document-level pairs due to computational constraints and limit this experiment to same-sentence pairs (see Appendix S6.2). For E3C, we compare SPANTREX and GRAPHTREX against multi-head attention and SPERT.³

5.3 Results and Discussion

Table 1 shows that GRAPHTREX sets a new state-of-the-art in end-to-end TREX on I2B2 2012, with a 68.81% tempeval F_1 , a 5.45% gain over Xu et al. (2013). GRAPHTREX also outperforms SPANTREX with more than 2% improvement due to an improved recall. The improvement in document-level F_1 scores is also highly statistically significant (p-value = 4.26×10^{-9} , paired t-test). Even with added model complexity, these gains are crucial for the clinical domain and patient outcomes. The multi-head attention method has the lowest performance among all baselines despite achieving state-of-the-art on the THYME corpus. While multi-head attention between two entities can effectively capture word-level depen-

dencies, it may still struggle with long-range dependencies without additional global or local context. In contrast, SPANTREX incorporates additional local context. GRAPHTREX further brings node-specific and edge-specific multi-head attention and provides additional global context, allowing interactions between existing high-confidence relations to improve the inference over long-range relations too. Additionally, the pipeline approach, limited to same-sentence relations, struggles with the NO-RELATION class, achieving only 10.79% F_1 . In contrast, GRAPHTREX effectively distinguishes relevant relations from the majority of negative samples, achieving 59.67% F_1 in sentence-level pairs. We also train our model for relation extraction with gold entities and find that the model severely underperforms, showing the importance of joint training with shared representations.

Qualitative Analysis We conduct an analysis of the system outputs against the ground truth and present a case study in Appendix S7, Figure S3. The example reveals an inconsistency in the ground truth, where ‘discharge’ event overlaps with both the admission and discharge dates. While SPANTREX replicates this error, GRAPHTREX does not, showing greater robustness. GRAPHTREX output is also more balanced across relation types and has denser connections than the ground truth, while SPANTREX identifies fewer *Before/After* pairs. SPANTREX misses two important wellness indicators—‘tolerated’ and ‘treatment’. Beyond this case study, we also find that the reported entity extraction scores may not fully reflect our model’s capability, as it identifies meaningful entities even when the gold standard omits them. For example, given {‘admission’, ‘hospitalization’}, GRAPHTREX and SPANTREX correctly identify both while the ground truth labels only the first.

²To ensure a fair comparison, we use the same entity-extraction setup as SPANTREX and implement multi-head attention over an $n \times n$ matrix of entities for relation decoder.

³We exclude baselines from 2013 due to code unavailability and additional private annotations, and REBEL and pipeline approaches due to poor performance on I2B2 2012.

Generalization to E3C Corpus We consider EVENTS and their TLinks since E3C data is sparsely annotated and majority (85.63%) of annotated TLinks occur between EVENTS only (Appendix S2.2). Table 2 shows SPANTREX outperforms SPERT by a large margin and GRAPHTREX gives further 1% improvement, showing strong generalizability of the method on this challenging dataset. We also attain improved entity extraction scores compared to the previous best of 63.44% (Zanoli et al., 2024).

Model	EVENT	TLink
	F_1	F_1
Multi-head Attention	84.0	3.97
SPERT	78.85	13.63
SPANTREX (ours)	81.3	22.55
GRAPHTREX (ours)	82.10	23.48

Table 2: Results on the E3C Corpus.

5.4 Ablation Analyses

Here we present further detailed analysis and ablation studies on the I2B2 corpus and provide additional experiments in Appendix S6.

Distance (d_r)	\bar{n}_r	SPANTREX			GRAPHTREX			%IMP
		P	R	F_1	P	R	F_1	
$d_r = 0$	159.4	75.4	41.1	53.2	72.4	45.1	55.6	4.5
$d_r > 0$	70.6	93.6	15.6	26.8	91.0	17.0	28.6	6.7
$d_r > 1$	35.9	94.4	8.6	15.7	92.8	9.4	17.1	8.9

Table 3: TLink scores by entity pair distance on I2B2: same window ($d = 0$), across windows ($d > 0, d > 1$). \bar{n}_r is the average ground truth TLinks. %IMP measures F_1 gain of GRAPHTREX over SPANTREX.

Does the GNN module improve long-range predictions? We investigate whether local versus global information exchange across an HG improves performance for lexically distant entities. We compare the TLink predictions over the same context window ($d_r = 0$) for local TLinks, across different windows ($d_r > 0$), and farther than adjacent windows ($d_r > 1$) for global TLinks. Here d_r denotes the number of windows between two entities. Table 3 shows GRAPHTREX outperforms SPANTREX across all distances, with a % improvement (lift) of 4.5% for same-window relations, 6.7% for those spanning one or more windows, and 8.9% for more distant pairs.

An important finding of this experiment is the scalability potential of GRAPHTREX as it improves

structural dependencies across long documents with graphs while keeping the relations and context invariant to the document length. This is an important advantage since the longer a document gets, the harder it is for a text-based model or transformer LMs to process it and capture long-range dependencies (Yuan et al., 2023; Qin et al., 2023a).

Variations in Pair Embedding Approaches. To enhance SPANTREX, we include (1) additional context by pooling all the tokens between entity spans and (2) predicted entity types in the pair embeddings ($\epsilon_{i,j}$ (Eq. 5)). (3) We also augment training data by adding inverse flipped relations—if $A = B$, we add $B = A$, if $A < B$, we add $B > A$ and vice-versa. Figure 4 shows the percentage improvement w.r.t. the SPANTREX, with the highest boost from additional context. Additional flipped relations also provide considerable gains while adding entity types leads to marginal gains. Even on the E3C corpus, additional context is crucial, as performance drops to 8.44% without it, and without flipping the *Overlap* relation, it drops to 20.95%.⁴

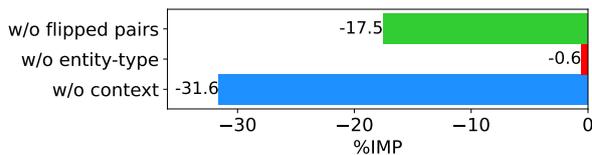


Figure 4: % F_1 gain on I2B2 of SPANTREX over ablations omitting specific pair representation components.

Ablation on Graph Construction/Impact of Node Types. We analyze the impact of each HG component by varying the node types used for graph construction and compare the performance with SPANTREX (Table 4). Using only entity nodes can degrade performance, likely due to the entity nodes overfitting their local neighborhoods in the absence of a broader context. Adding context nodes improves results, consistent with Figure 4, showing the importance of context between spans. Including both the context nodes and the window nodes further enhances the model performance, showing that they are mutually compatible.⁵

Variations Across Different Encoders We experiment with several other base encoders and report the results with two popular models—BERT and RoBERTa—in Table 5. We find that GRAPH-

⁴Before pairs are not flipped in E3C since it doesn't have the inverse *After* class.

⁵Ablation on E3C corpus gives slightly different trends, given the short documents (Appendix S6.4).

Method	Graph Components			P	R	F	%IMP
	Span	Context	Window				
SPANTREX				81.33	56.44	66.63	-
GRAPHTREX	✓	✗	✗	80.26	53.67	64.43	-3.3
	✓	✓	✗	80.34	58.75	67.87	1.9
	✓	✗	✓	77.23	59.98	67.52	1.3
	✓	✓	✓	78.21	61.42	68.81	2.3

Table 4: % F_1 gains (IMP) of GRAPHTREX variants over SPANTREX for the I2B2 dataset.

TREX provides additional gains over SPANTREX across both encoders. We also report the results with clinical Longformer (Li et al., 2022), designed to process long clinical texts. In line with Qin et al. (2023b) we find that in their native implementation using SPANTREX, the transformer-based models including Longformer struggle with long-range dependencies. Clinical Longformer also underperforms over GRAPHTREX with F_1 score 63.84%. We also report the results on entity extraction with several pre-trained clinical base encoders in Appendix S6.1 Table S6, and find that none of them outperform BioMedBERT.

Model	TLink (end-to-end)		
	P	R	F_1
GRAPHTREX	78.21	61.42	68.81
BERT-SPANTREX	75.94	48.63	59.30
BERT-GRAPHTREX	77.45	52.30	62.44
RoBERTA-SPANTREX	79.98	52.05	63.06
RoBERTa-GRAPHTREX	78.22	54.88	64.50
Clinical-Longformer	82.39	52.10	63.84

Table 5: Variations across encoders on the I2B2 data.

Performance Across Relation Classes We report the performance of GRAPHTREX across individual TLink classes in Table 6. While reporting these scores for E3C is straightforward since the evaluation metric is micro F_1 , for I2B2, we report tempeval metrics. Therefore, before computing these metrics, we filter the system and gold output to the desired relation type. Note that during training, we enrich the training data with corresponding inverse relations by flipping the participant entity order, due to which the model learns to output an equivalent number of *Before/After* relations. However, for the gold standard, there are fewer TLinks for *After*. Therefore, to have a fair evaluation, we similarly flip the relation type in the gold standard before computing the above scores. The number

of TLinks reported in the above table is based on the I2B2 evaluation script; they represent the total compatible links in the gold standard test set, given the match in extracted entities. As seen from this table, the model performance is better on the majority class for both datasets.

Corpus	RelType	#TLinks	F1
I2B2 2012	<i>Overlap</i>	7155	59.27
	<i>Before</i>	13453	75.05
	<i>After</i>	13453	74.94
E3C	<i>Overlap</i>	1560	24.23
	<i>Before</i>	793	21.88

Table 6: Performance Across Relation Classes.

6 Conclusion and Future Work

We introduce GRAPHTREX, a novel model for temporal relation extraction in long clinical documents. By integrating span-based extraction with GNNs, GRAPHTREX provides a structured and context-aware framework, achieving a new state-of-the-art in end-to-end TREX on the I2B2 2012 corpus. Our ablation studies show that constructing a document-level heterogeneous graph with special landmark nodes more effectively captures local and global information, enhancing long-range relation extraction. Experiments on the E3C corpus demonstrate the model’s generalizability beyond the U.S. context. We also introduce BioMedBERT-UMLS, a knowledge-infused entity extraction model achieving state-of-the-art in TimEx extraction. Due to non-trivial scalability challenges, we consider knowledge integration for TREX a future direction.

Accurate event chronology is crucial for clinical applications, including opportunistic screening for chronic diseases like Type 2 Diabetes, which affects millions worldwide (IDF, 2025) and is a key focus of our ongoing research. This is especially vital for marginalized communities lacking regular access to care. Clinical notes from even a single patient visit capture richer historical context than structured EHR but remain underutilized. By enabling more precise TREX from these narratives, GRAPHTREX lays the foundation for enhanced event sequence modeling for health risk prediction, treatment optimization, clinical decision support, and patient education. Integrating such systems with human-in-the-loop approaches can mitigate missed diagnoses, facilitate opportunistic screening, and advance healthcare for all.

Limitations

One of the limitations of the current work is the constraint to validate the model across more clinical datasets due to their unavailability. In the clinical domain, concerns about the leakage of protected health information limit data release and cross-hospital annotation efforts. The only other moderate-scale dataset THYME (Bethard et al., 2017; Styler IV et al., 2014), requires a membership fee similar to the Linguistic Data Consortium (LDC). However, we show generalizability to another recent small-scaled dataset. Due to the ease of implementation, code availability, and adaptability of our approach, it can be extended for the future works once additional datasets become available or accessible.

More broadly, GRAPHTREX enables downstream research in healthcare and clinical decision support, rather than as a standalone diagnostic tool. In real-world applications, clinical NLP models like ours are typically integrated into a broader analytics ecosystem, for example, in Type 2 Diabetes opportunistic screening research in our own research ecosystem, where we are applying it to multiple note types beyond just discharge summaries and applying it for multi-document temporal reasoning. While strong model performance has positive societal impacts—such as improving early risk detection, optimizing treatment pathways, reducing clinician burden, and improving patient education with summarized timelines—these systems should be used with human oversight to manage the risks associated with model limitations. Future research should focus on robust evaluation across diverse clinical settings, to ensure reliable, and contextually appropriate deployment of the system.

Ethics Statement

This research uses de-identified clinical notes from Beth Israel Medical Center in Boston. The data was released by Harvard Medical School and is available with necessary ethical approvals. We have signed a Data Use Agreement (DUA) with n2c2 to ensure compliance with privacy regulations and safeguard data confidentiality. We take all necessary precautions to prevent misuse of the data and unauthorized access to the data. Additionally, we also show experimental results on the E3C corpus constructed from publicly available clinical documents. Our work aims to enhance clinical decision-making by extracting temporal relations

from medical events, supporting healthcare professionals' judgment without undermining it. By accurately extracting detailed information from clinical notes, which capture the nuances of patient history, GRAPHTREX can provide a more comprehensive and accurate summary of patient timelines. This capability helps bridge diagnostic disparities, particularly for marginalized communities with irregular access to care or incomplete medical records, ensuring healthcare providers have a more complete view of a patient's medical journey. Given the potential impact on patient outcomes, ongoing validation and evaluation are essential to ensure that the model's performance is robust, accurate, and equitable, mitigating risks of misinterpretation and unintended harm.

Acknowledgment

We thank the anonymous reviewers for their valuable feedback. This work is supported by NSF IIS #2312862.

References

2025. *IDF Diabetes Atlas*, 11 edition. International Diabetes Federation, Brussels, Belgium.
- Ghada Alfattni, Niels Peek, and Goran Nenadic. 2020. Extraction of temporal relations from clinical free text: A systematic review of current approaches. *Journal of biomedical informatics*, 108:103488.
- James F Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.
- Emily Alsentzer, John R Murphy, Willie Boag, Weihung Weng, Di Jin, Tristan Naumann, WA Redmond, and Matthew BA McDermott. 2019. Publicly available clinical bert embeddings. *NAACL HLT 2019*, page 72.
- Alan R Aronson, Thomas C Rindfleisch, and Allen C Browne. 1994. Exploiting a large thesaurus for information retrieval. In *RIAO*, volume 94, pages 197–216.
- Peyman Baghershahi, Reshad Hosseini, and Hadi Moradi. 2023. [Efficient relation-aware neighborhood aggregation in graph neural networks via tensor decomposition](#). *Preprint*, arXiv:2212.05581.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Steven Bethard, Guergana Savova, Martha Palmer, and James Pustejovsky. 2017. [SemEval-2017 task 12: Clinical TempEval](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 565–572, Vancouver, Canada. Association for Computational Linguistics.
- Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.
- Daniel Capurro, Erik van Eaton, Robert Black, and Peter Tarczy-Hornoch. 2014. Availability of structured and unstructured clinical data for comparative effectiveness research and quality improvement: a multisite assessment. *EGEMS*, 2(1).
- Chunkit Chan, Jiayang Cheng, Weiqi Wang, Yuxin Jiang, Tianqing Fang, Xin Liu, and Yangqiu Song. 2023. Chatgpt evaluation on sentence level relations: A focus on temporal, causal, and discourse relations. *arXiv preprint arXiv:2304.14827*.
- Rochana Chaturvedi, Mudassir Rashid, Brian T Layden, Andrew Boyd, Ali Cinar, and Barbara Di Eugenio. 2023. Sequential representation of sparse heterogeneous data for diabetes risk prediction. In *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 831–834. IEEE.
- Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Cheng Cheng and Jeremy C Weiss. 2023. Typed markers and context for clinical temporal relation extraction. In *Machine Learning for Healthcare Conference*, pages 94–109. PMLR.
- Yang Cui, Lifeng Han, and Goran Nenadic. 2023. Medtem2. 0: Prompt-based temporal classification of treatment events from discharge summaries. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 160–183.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Teerapon Dhippayom, Anjana Fuangchan, Sirirat Tunpichart, and Nathorn Chaiyakunapruk. 2013. Opportunistic screening and health promotion for type 2 diabetes: an expanding public health role for the community pharmacist. *Journal of Public Health*, 35(2):262–269.
- Kalpit Dixit and Yaser Al-Onaizan. 2019. Span-level model for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5308–5314.
- Dmitriy Dligach, Timothy Miller, Chen Lin, Steven Bethard, and Guergana Savova. 2017. Neural temporal relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 746–751.
- Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. In *ECAI 2020*, pages 2006–2013. IOS Press.
- Jinlan Fu, Pengfei Liu, and Graham Neubig. 2020. Interpretable multi-dataset evaluation for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6058–6069.
- Jun Gao, Huan Zhao, Changlong Yu, and Ruifeng Xu. 2023. Exploring the feasibility of chatgpt for event extraction. *arXiv preprint arXiv:2303.03836*.
- Marzyeh Ghassemi, Marco A. F. Pimentel, Tristan Naumann, Thomas Brennan, David A. Clifton, Peter Szolovits, and Mengling Feng. 2015. A multivariate timeseries modeling approach to severity of illness assessment and forecasting in icu with sparse,

- heterogeneous clinical data. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 446–453. AAAI Press.
- Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.
- 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)*, pages 434–444.
- Rujun Han, Yichao Zhou, and Nanyun Peng. 2020. Domain knowledge empowered structured neural net for end-to-end event temporal relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5717–5729.
- William R Hersh, Mark G Weiner, Peter J Embi, Judith R Logan, Philip RO Payne, Elmer V Bernstam, Harold P Lehmann, George Hripcsak, Timothy H Hartzog, James J Cimino, et al. 2013. Caveats for the use of operational electronic health record data in comparative effectiveness research. *Medical care*, 51:S30–S37.
- Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*, page 2704–2710.
- Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. **REBEL: Relation extraction by end-to-end language generation**. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Tuan Lai, Heng Ji, ChengXiang Zhai, and Quan Hung Tran. 2021. Joint biomedical entity and relation extraction with knowledge-enhanced collective inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6248–6260.
- Dongha Lee, Xiaoqian Jiang, and Hwanjo Yu. 2020. Harmonized representation learning on dynamic ehr graphs. *Journal of biomedical informatics*, 106:103426.
- Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023a. Evaluating chatgpt’s information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *arXiv preprint arXiv:2304.11633*.
- Jing Li, Yequan Wang, Shuai Zhang, and Min Zhang. 2023b. **Rethinking document-level relation extraction: A reality check**. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5715–5730, Toronto, Canada. Association for Computational Linguistics.
- Manling Li, Sha Li, Zhenhailong Wang, Lifu Huang, Kyunghyun Cho, Heng Ji, Jiawei Han, and Clare Voss. 2021. The future is not one-dimensional: Complex event schema induction by graph modeling for event prediction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5203–5215.
- Yikuan Li, Ramsey M Wehbe, Faraz S Ahmad, Hanyin Wang, and Yuan Luo. 2022. Clinical-longformer and clinical-bigbird: Transformers for long clinical sequences. *arXiv preprint arXiv:2201.11838*.
- Chae-Gyun Lim, Young-Seob Jeong, and Ho-Jin Choi. 2019. Survey of temporal information extraction. *Journal of Information Processing Systems*, 15(4):931–956.
- Chen Lin, Dmitriy Dligach, Timothy A Miller, Steven Bethard, and Guergana K Savova. 2016a. Multilayered temporal modeling for the clinical domain. *Journal of the American Medical Informatics Association*, 23(2):387–395.
- Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2016b. Improving temporal relation extraction with training instance augmentation. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 108–113.
- Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2019. A bert-based universal model for both within-and cross-sentence clinical temporal relation extraction. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 65–71.
- Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2021. **EntityBERT: Entity-centric masking strategy for model pretraining for the clinical domain**. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 191–201, Online. Association for Computational Linguistics.
- Jian Liu, Jinan Xu, Yufeng Chen, and Yujie Zhang. 2021. Discourse-level event temporal ordering with uncertainty-guided graph completion. In *IJCAI*, pages 3871–3877.
- Sijia Liu, Liwei Wang, Vipin Chaudhary, and Hongfang Liu. 2019. **Attention neural model for temporal relation extraction**. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 134–139, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

- Zengjian Liu, Ming Yang, Xiaolong Wang, Qingcai Chen, Buzhou Tang, Zhe Wang, and Hua Xu. 2017. Entity recognition from clinical texts via recurrent neural network. *BMC medical informatics and decision making*, 17:53–61.
- Bernardo Magnini, Begoña Altuna, Alberto Lavelli, Anne-Lyse Minard, Manuela Speranza, and Roberto Zanolini. 2022. European clinical case corpus. In *European Language Grid: A Language Technology Platform for Multilingual Europe*, pages 283–288. Springer International Publishing Cham.
- Ramon Maldonado, Meliha Yetisgen, and Sanda M Harabagiu. 2019. Adversarial learning of knowledge embeddings for the unified medical language system. *AMIA Summits on Translational Science Proceedings*, 2019:543.
- 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.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.
- Timothy Miller, Steven Bethard, Dmitriy Dligach, and Guergana Savova. 2023. End-to-end clinical temporal information extraction with multi-head attention. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 313–319.
- Qiang Ning, Zhili Feng, and Dan Roth. 2017. A structured learning approach to temporal relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1027–1037.
- Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65.
- Perry J Pickhardt, Peter M Graffy, Alberto A Perez, Meghan G Lubner, Daniel C Elton, and Ronald M Summers. 2021. Opportunistic screening at abdominal ct: use of automated body composition biomarkers for added cardiometabolic value. *RadioGraphics*, 41(2):524–542.
- Guanghui Qin, Yukun Feng, and Benjamin Van Durme. 2023a. [The NLP task effectiveness of long-range transformers](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3774–3790, Dubrovnik, Croatia. Association for Computational Linguistics.
- Guanghui Qin, Yukun Feng, and Benjamin Van Durme. 2023b. The nlp task effectiveness of long-range transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3774–3790.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. [Improving language understanding by generative pre-training](#).
- Omid Rohanian, Mohammadmahdi Nouriborji, and David A Clifton. 2023. Exploring the effectiveness of instruction tuning in biomedical language processing. *arXiv preprint arXiv:2401.00579*.
- José Javier Saiz and Begoña Altuna. 2023. End-to-end temporal relation extraction in the clinical domain. In *Text2Story@ ECIR*.
- Jane Scheetz, Dilara Koca, Myra McGuinness, Edith Holloway, Zachary Tan, Zhuoting Zhu, Rod O’Day, Sukhpal Sandhu, Richard J MacIsaac, Chris Gilfillan, et al. 2021. Real-world artificial intelligence-based opportunistic screening for diabetic retinopathy in endocrinology and indigenous healthcare settings in Australia. *Scientific Reports*, 11(1):1–11.
- Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and Philip S. Yu. 2017. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):17–37.
- Yuqi Si, Jingqi Wang, Hua Xu, and Kirk Roberts. 2019. Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association*, 26(11):1297–1304.
- Sunghwan Sohn, Kavishwar B Waghlikar, Dingcheng Li, Siddhartha R Jonnalagadda, Cui Tao, Ravikumar Komandur Elayavilli, and Hongfang Liu. 2013. Comprehensive temporal information detection from clinical text: medical events, time, and tlink identification. *Journal of the American Medical Informatics Association*, 20(5):836–842.
- William F Styler IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C De Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, et al. 2014. Temporal annotation in the clinical domain. *Transactions of the association for computational linguistics*, 2:143–154.
- Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013a. Annotating temporal information in clinical narratives. *Journal of biomedical informatics*, 46:S5–S12.
- Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013b. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association: JAMIA*, 20(5):806.
- Buzhou Tang, Yonghui Wu, Min Jiang, Yukun Chen, Joshua C Denny, and Hua Xu. 2013. A hybrid system for temporal information extraction from clinical text. *Journal of the American Medical Informatics Association*, 20(5):828–835.

- Julien Tourille, Olivier Ferret, Aurelie Neveol, and Xavier Tannier. 2017. Neural architecture for temporal relation extraction: A bi- lstm approach for detecting narrative containers. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 224–230.
- Nastassia Tvardik, Ivan Kergourlay, André Bittar, Frédérique Segond, Stefan Darmoni, and Marie-Hélène Metzger. 2018. Accuracy of using natural language processing methods for identifying healthcare-associated infections. *International Journal of Medical Informatics*, 117:96–102.
- Naushad UzZaman and James Allen. 2011. Temporal evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 351–356.
- Siddharth Vashishtha, Adam Poliak, Yash Kumar Lal, Benjamin Van Durme, and Aaron Steven White. 2020. Temporal reasoning in natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4070–4078.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.
- Liang Wang, Peifeng Li, and Sheng Xu. 2022. Dct-centered temporal relation extraction. In *Proceedings of the 29th international conference on computational linguistics*, pages 2087–2097.
- Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The World Wide Web Conference*, page 2022–2032.
- Yan Xu, Yining Wang, Tianren Liu, Junichi Tsujii, and Eric I-Chao Chang. 2013. An end-to-end system to identify temporal relation in discharge summaries: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):849–858.
- Zhaohui Yan, Zixia Jia, and Kewei Tu. 2022. An empirical study of pipeline vs. joint approaches to entity and relation extraction. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 437–443.
- Xi Yang, Jiang Bian, William R Hogan, and Yonghui Wu. 2020. Clinical concept extraction using transformers. *Journal of the American Medical Informatics Association*, 27(12):1935–1942.
- Jiarui Yao, Harry Hochheiser, WonJin Yoon, Eli Goldner, and Guergana Savova. 2024. [Overview of the 2024 shared task on chemotherapy treatment timeline extraction](#). In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 557–569, Mexico City, Mexico. Association for Computational Linguistics.
- Jie Yu, Bin Ji, Shasha Li, Jun Ma, Huijun Liu, and Hao Xu. 2022. S-ner: A concise and efficient span-based model for named entity recognition. *Sensors*, 22(8):2852.
- Chenhan Yuan, Qianqian Xie, and Sophia Ananiadou. 2023. [Zero-shot temporal relation extraction with ChatGPT](#). In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 92–102, Toronto, Canada. Association for Computational Linguistics.
- Roberto Zanolli, Alberto Lavelli, Daniel Verdi do Amarante, and Daniele Toti. 2024. Assessment of the e3c corpus for the recognition of disorders in clinical texts. *Natural Language Engineering*, 30(4):851–869.
- Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 793–803.
- Shuaicheng Zhang, Lifu Huang, and Qiang Ning. 2021. Extracting temporal event relation with syntax-guided graph transformer. *arXiv preprint arXiv:2104.09570*.
- Yurong Zhang, Gang Hu, Lu Zhang, Rachel Mayo, and Liwei Chen. 2015. A novel testing model for opportunistic screening of pre-diabetes and diabetes among us adults. *PLoS One*, 10(3):e0120382.
- Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61.
- Jie Zhou, Shenpo Dong, Hongkui Tu, Xiaodong Wang, and Yong Dou. 2022. Rsgt: relational structure guided temporal relation extraction. In *Proceedings of the 29th international conference on computational linguistics*, pages 2001–2010.
- Yichao Zhou, Yu Yan, Rujun Han, J Harry Caufield, Kai-Wei Chang, Yizhou Sun, Peipei Ping, and Wei Wang. 2021. Clinical temporal relation extraction with probabilistic soft logic regularization and global inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 14647–14655.

APPENDIX

S1 Additional Related Literature

Temporal Information Extraction. The entity and relation extraction techniques have evolved from machine learning approaches such as Naive Bayes, Markov Logic Network, SVM, CRF, and Structured Perceptron to deep neural networks and recently large pre-trained language models and graph neural networks (Lim et al., 2019; Alfatni et al., 2020). The features for training these systems range from rule-based, lexical (uppercase, stopword, sentence lengths, etc.), syntactic and semantic (part-of-speech, tense, dependency paths, etc.), ontology-based (for instance, the top ontology class from Wordnet), to distributional features (word embeddings, pre-trained language model embeddings). For relation extraction, while some works include candidate pair identification and classification both (RE) (Xu et al., 2013; Tang et al., 2013), others focus on relation classification (RC), given the candidate pairs (Zhou et al., 2021). While some develop separate classifiers for E-E (event-event), E-T (event-timex), and T-T (timex-timex) relations (Tang et al., 2013; Lin et al., 2016a), others model different subtypes together using the same classifier (Zhou et al., 2021). (Han et al., 2020) focus on a sub-problem of end-to-end relation extraction where they only consider E-E (event-event) pairs across short neighbourhoods (3 sentence chunks). Pre-trained word embeddings with additional manually engineered features have been utilized with LSTM (Tourille et al., 2017; Liu et al., 2019) and CNN (Dligach et al., 2017) networks on the THYME dataset. Subsequently, the introduction of attention mechanism and transformer-based pre-trained language models (PLM) like BERT offered further significant improvements on both entities (Alsentzer et al., 2019; Si et al., 2019; Yang et al., 2020) and relations (Lin et al., 2019; Zhou et al., 2021). The methods for relation classification involve inserting special marker tokens to better represent entity boundaries and processing the sequence using a BERT-based model. This requires creation of such sequences for every entity pair, therefore, at the document level, the task complexity becomes too onerous if the negative samples, i.e. the pairs with No_Relation are also included. These works therefore remain restricted to the simpler task of relation classification, which excludes the majority No_Relation class.

Relation Extraction using Additional Knowledge. Some of the works also leverage additional knowledge such as temporal calculus (Allen, 1983) to either infer additional relations or improve consistency of the identified relations (Ning et al., 2017). Zhou et al. (2021) address the relation classification problem and incorporate temporal calculus using probabilistic soft logic (PSL) rules to improve over a simple BERT-based baseline by a significant margin. Others incorporate clinical domain knowledge, such as Xu et al. (2013), who use additional private and public annotations to improve entity extraction, Lin et al. (2016b), who enrich the training set using Unified Medical Language System (UMLS) ontology (Bodenreider, 2004) and demonstrate improvements on the THYME dataset, others model the data-driven constraints (Han et al., 2019, 2020). More recently, Li et al. (2023b) leverages the neighboring tokens of each event as local cues and the temporal words between the two events as the global cues to improve the relation classification performance in the general domain. Most of the recent works either perform relation classification given the gold entity pairs or only report performance on a subset of relations such as event-event relations over 1-3 adjacent sentences and ignore the long-distance relations.

Relation Extraction using LLMs. Large language models (LLMs) such as BART (Saiz and Altuna, 2023), T5 (Cui et al., 2023), and Llama2 (Rohanian et al., 2023) have also been applied to TREX. However, at present, these models perform significantly worse than the large specialized pre-trained model-based approaches within both clinical (Saiz and Altuna, 2023) and general domain TREX (Gao et al., 2023; Li et al., 2023a), more so for long-range temporal relations (Yuan et al., 2023; Chan et al., 2023).

Relation Extraction using GNNs. GNNs have been widely applied to heterogeneous graphs, a.k.a knowledge graphs for different applications (Zhang et al., 2019; Wang et al., 2019; Baghersahi et al., 2023), particularly relation extraction. Prior works use GNNs with document-level graphs using syntactic, semantic, and discourse features for temporal relation classification, given the entities (Mathur et al., 2021; Zhou et al., 2022), or for relation extraction over 1-2 sentences (Zhang et al., 2021). Wang et al. (2022) anchor similarly constructed graphs using the document creation time and obtain considerable improvements. Liu et al. (2021)

train a GNN-based model to recover masked edges during training and use the most confident predictions to recover the edges during inference. However, none of these works focus on document-level end-to-end TREX or the clinical domain.

S2 Data

S2.1 I2B2 2012 Corpus

Pre-processing. For the I2B2 2012 corpus, there were several errors and omissions in the gold annotations that were addressed during preprocessing. To begin with, a large number of character offsets required correction when reading from the XML files. Some entity types were not annotated, we omitted these and also the relations involving such omitted entities. Some relations were missing head/tail, these were omitted as well. In several documents, DISCHARGE was not annotated as a SECTIME and still referred to in the TLinks with id “Discharge”—this was replaced with the id of an entity with text “Discharge” if available, otherwise these relations were omitted as well.

Additional Data Description. We show the distributions of the number of tokens, the number of entities, and the number of relations in each file in the test split in Figure S1. On average, each file has 786 tokens, 128 annotated entities, and 229 annotated relation pairs. The maximum token length across the test set is 2503, the maximum number of annotated entities is 344, and the highest number of relations in any file is 639. Note that although relation pairs can be quadratic in terms of number of entities, the annotated relation pairs in the I2B2 dataset are sparse—annotation density is 21% (Zhou et al., 2021)—due to the difficulty of annotating the long documents. Therefore, to address this, the closure computations are included during the evaluation.

Entity Description. The span-based entity-extraction models start by enumerating all possible contiguous spans in the document. For example, in a document with n tokens $d = [t_1, t_2, t_3, \dots, t_n]$, the model produces a list of $O(n^2)$ spans. Classifying all possible spans not only increases the computational burden but also increases the data imbalance by introducing many more Not-Entity types, since usually the entity lengths have a much lower bound compared to a document’s length. A common way of handling this issue is to place an upper bound on the length of the possible entity spans. We choose

a bound of 7 tokens which falls under the 97th percentile of all entity lengths in the training set. We show the percentile of 7 across each entity type in the training set in Table S1. We also show the distributions of token lengths across each entity type in Figure S2 using kernel density plots as they give a continuous and smooth curve estimating the probability density function. Note that on average, all entity types are shorter than 7 tokens. The appearance of negative token lengths in the distributions in Figure S2 result of the KDE smoothing process. Only a few instances of type PROBLEM, TEST, TREATMENT, or CLINICAL DEPARTMENT can be longer. For example, the longest token length (using BioMedBERT tokenizer) in the test set for TIME entity is 14 in ‘5-2-98 , at 4:55 p.m.’, for FREQUENCY, it is 11 in ‘Q4-6H (every 4 to 6 hours)’, and for Clinical_Department, it is 18, and includes a department, physician, and clinic names. We also increased the maximum span length to 20. However, it did not improve the results.

There are several boundary overlaps among the entities, including overlaps in the same types (e.g., 10/9/2024 vs. 10/9, both annotated as DATE). Therefore, when compiling the gold set, we prioritize the first annotation. We also experiment with choosing the largest entity span as a tie-breaker strategy for the overlapping spans for training purposes. However, this reduces the performance. On average the entity annotations in the dataset have high label consistency (97.68% in the training set and 98.13% in the test set). Entity label consistency is an important measure to validate the agreement in the annotation. Specifically, the span-level entity label consistency ϕ_{eCon} computes whether the same sequence of tokens appearing multiple times in the dataset are annotated with the same entity type, and has a significant impact on an entity classification model’s performance (Fu et al., 2020). $\phi_{eCon}(e_{sp}) = |y_{sp} == E_{sp}|/|e_{sp}|$, where E_{sp} is the most frequently annotated class for entity span e_{sp} . We report the span-level entity consistency ϕ_{eCon} of individual entity classes in Table S1. Note that the label consistency of ADMISSION and DISCHARGE is low and that for DATE is relatively lower than other entity types. Upon further analysis, we find that the ADMISSION/DISCHARGE entities always overlap with the DATE type. Therefore, to address this issue, we set a priority order to the entity types for label assignment for training purposes. For example, we set the highest priority to ADMISSION/DISCHARGE, then if an entity

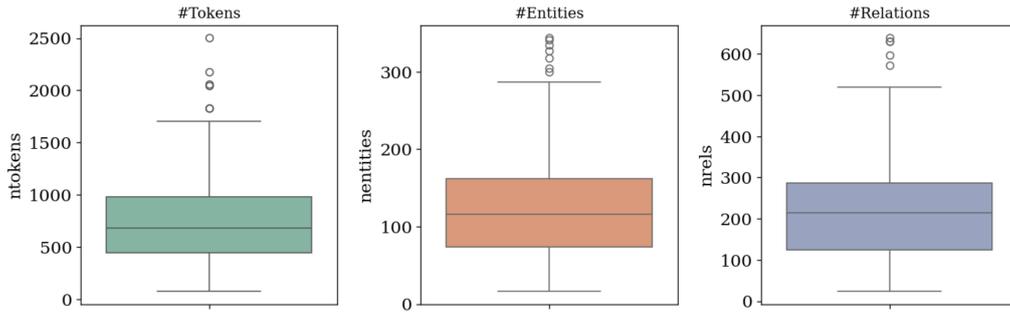


Figure S1: Distributions of the number of tokens, number of entities, and number of relations in each file over the test set in the I2B2 corpus.

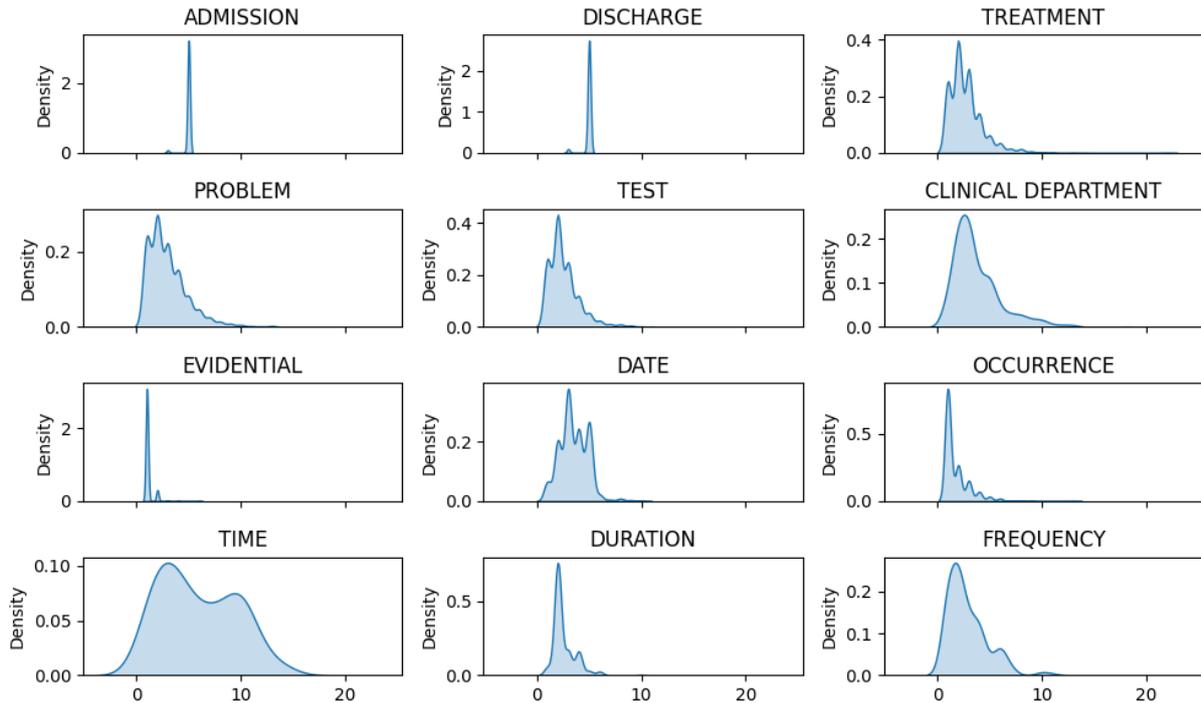


Figure S2: Distributions of the token lengths across each entity type in the test set of the I2B2 corpus.

is annotated with additional labels other than ADMISSION/DISCHARGE, we ignore it. There are numerous other occurrences for DATE type entities and the model can learn to identify DATE types from those. During inference, as an additional post-processing, we add additional DATE type entities for each ADMISSION/DISCHARGE entity.

Relation Description. We report the number of relation types across the training, development, and test splits in Table S2. The data has a high imbalance, with relatively fewer annotations for *After* relations. To address the imbalance during training we flip the relations pairs participating in *Before* to create additional pairs for *After*, and vice-versa. This creates an equal number of *Before*, and *After* relation types for training. However such augmen-

tation introduces some imbalance w.r.t. *Overlap* relation, even after augmenting additional *Overlap* relations by flipping the *Overlap* pairs.

S2.2 E3C Corpus

We download the corpus version 2.0.0 from <https://live.european-language-grid.eu/catalogue/corpus/7618/download/>. We only use the English subset and the first layer comprising manual annotations. We then select a 20% random sample of documents from the training set to form the development set and save the best-performing model on this set.

Pre-processing. We pre-process the data to adjust the character offsets (primarily due to carriage returns), aligning the text spans with annotated en-

EType	Frequency	Consistency ϕ_{eCon} (%)	Percentile (7)
PROBLEM	4814	99.2	94.8
TREATMENT	3681	98.7	97.2
TEST	2507	98.7	98.2
CLINICAL DEPARTMENT	957	97.9	90.0
EVIDENTIAL	714	90.4	100.0
OCCURRENCE	3122	95.7	99.1
ADMISSION	168	52.3	100.0
DISCHARGE	163	52.6	100.0
DATE	1222	85.0	98.9
TIME	68	97.0	55.9
DURATION	388	95.7	99.1
FREQUENCY	240	98.7	99.6

Table S1: Entity types description for the training set of the I2B2 corpus. Average label consistency over the training set is 97.68, and the percentile of 7 across all entity types is 97.

Relation Type	Train	Dev	Test
<i>#Overlap</i>	12278	538	9889
<i>#Before</i>	16657	642	14920
<i>#After</i>	3075	127	2724

Table S2: Number of annotated relations across the training, development, and test set in the I2B2 corpus.

tity names.

E3C Entity Description. The dataset has following entity types BODYPART, PATIENT, EVENT, DATE, OTHER and timexes including PREPOST-EXP, DURATION, H-PROFESSIONAL, QUANTIFIER, SET. However, there are only a few TLinks involving at least one non-EVENT entity (14.37% in the test set). In contrast, 85.63% of TLinks are EVENT-EVENT links. Therefore, we only consider EVENT-EVENT TLinks in this work. On average, each EVENT in the training split comprises 1.2 tokens. However, the max token length can be 10. We use the same max span length as I2B2 (7) in our experiments which falls under 99.94th percentile of training data distribution.

E3C Relation Description. The dataset contains both doctime relations and pairwise relations between entities. We only focus on the latter in this work. This subset includes the following relation types: *Before*, *Contains*, *Overlap*, *Begins-On*, *Ends-On*, *Simultaneous*. Of these *Before* and *Contains* are the most frequent, similar to the I2B2 dataset, we merge the less frequent classes *Overlap*, *Simultaneous*, with *Contains*, and for clarity, denote it as the *Overlap* class. We also merge *Ends-On* with *Before*, and *Begins-On* with *Before* after flipping

Category	Train	Dev	Test
Document Statistics			
<i>Total Documents</i>	29	7	48
<i>Average Tokens</i>	406.14	350.43	392.67
<i>Max Tokens</i>	797	694	828
<i>Min Tokens</i>	118	129	122
Event Statistics			
<i>Total events</i>	1657	385	2843
<i>Max Token Length</i>	1.20	1.22	1.24
Relation Statistics			
<i>#Overlap</i>	962	220	1560
<i>#Before</i>	484	102	793
<i>%E-E pairs</i>	84.81	80.90	85.63

Table S3: Descriptive Statistics for the E3C Corpus.

the entity order.

Descriptive statistics of this dataset are reported in Table S3.

S3 Sliding Window Example

Given a window size of n (e.g. in BERT-based encoders, this can be at most 512), we use a stride of $n-2//2$ to enumerate overlapping windows. For example, consider the example in Table S4. Given a sequence with 8 tokens T_1, T_2, \dots, T_8 , and let the maximum window size be 6. Each window begins with a CLS and ends with a SEP token. Each token in the window attends to itself and three neighbors. We fix the value of $stride = 6 - 2//2 = 2$, i.e. the window moves 2 tokens at a time. While some tokens participate in overlapping windows, their final embedding is computed from a single window. In the other window, they only provide additional context for other tokens.

We clarify this process with reference to the example in Table S4. The goal is to ensure that each token receives a mask value of 1 exactly once across all windows. Here mask value 1 means the token contributes to embeddings, while -2 means the token embedding is used for context but it doesn't contribute to this token's final representation. Masking is performed in two steps:

1. Inner Window Construction:

- Each window has an *inner window size* = $\min(\text{window-size} - 2, \text{remaining-sequence-length})$. The last window may be shorter and is padded using [PAD] tokens to maintain

a consistent size. The stride is set to $\text{inner-window-size}/2$.

- **Initial Mask Assignment:** The first window assigns a Mask value of 1 to the leftmost eligible position, while subsequent windows initially assign 0s. The final window assigns a 1 to the rightmost eligible position, while others maintain 0s.
- Each window is then divided into three sections:
 - (a) **Left section:** $\text{inner} - \text{window} - \text{size}/4$ tokens, filled with the leftmost mask value (typically 0).
 - (b) **Middle section:** $\text{inner} - \text{window} - \text{size}/2$ tokens, filled with 1s (these are the active tokens contributing to final embeddings).
 - (c) **Right section:** Remaining positions, filled with the rightmost mask value (typically 0).

2. **Prefix and Suffix Application:** Special tokens [CLS], [SEP] are respectively added at the start and end of each window with their mask values: -3 . Any remaining positions are filled with [PAD] token (mask value = -4), which is ignored.

This process ensures that in the first window, tokens T_1, T_2, T_3 are actively covered (T_4 used for context but does not contribute to its own final representation). In the second window, T_4 and T_5 are covered, followed by T_6 and T_7 in the third, and in the fourth only T_8 is covered. Therefore, each token’s final representation is obtained from a unique window where it actively contributes.

Token Windows	Window Masks
[CLS, T₁ , T₂ , T₃ , T_4 , SEP]	[-3, 1, 1, 1, -2, -3]
[CLS, T_3 , T₄ , T₅ , T_6 , SEP]	[-3, -2, 1, 1, -2, -3]
[CLS, T_5 , T₆ , T₇ , T_8 , SEP]	[-3, -2, 1, 1, -2, -3]
[CLS, T_7 , T₈ , SEP, PAD, PAD]	[-3, -2, 1, -3, -4, -4]

Table S4: Illustration of sliding window construction for a sequence T_1, T_2, \dots, T_8 , with window size 6. Embeddings of active tokens (in boldface) are used while the others only provide additional context. In addition, the embeddings of CLS tokens in each window are also used to initialize the window nodes.

S4 Graph Construction

We use SPANTREX to construct our initial graph. We begin with mapping each predicted entity type to a node type in the graph. Then we filter all the relations for which the prediction probability $f(\epsilon_{i,j})$ of the relation decoder (f in Eq. 6) is above a pre-defined threshold τ . We model these high-confidence predictions as edges between the corresponding nodes. This thresholding is of significant importance because we train the GNN model end-to-end, and our goal is to avoid the propagation of noisy and spurious information through the graph.

Formally, given the predictions of g , we first construct the set of entity nodes $\mathcal{V}_{sp} = \{v_i^s \mid i \in [1 \dots N_v], \hat{y}_i^n \neq Y_{none}^n\}$, where $Y_{none}^n := \text{NOT-ENTITY}$, and its corresponding set of node types $\mathcal{A}_{sp} = \{\hat{y}_i^n \mid v_i^s \in \mathcal{V}_{sp}\}$. We also construct the set of edge types as $\mathcal{R}_{sp} = \{(\hat{y}_i^n, \hat{y}_{ij}^r, \hat{y}_j^n) \mid i, j \in [1 \dots N_v], i \neq j, \hat{y}_{ij}^r \neq Y_{none}^r, \hat{y}_i^n, \hat{y}_j^n \in \mathcal{A}_{sp}, \max \text{softmax}(f(\epsilon_{ij})) \geq \tau\}$, where $Y_{none}^r := \text{NO-RELATION}$. Additionally, we initialize the feature matrix of entity nodes as

$$X_{sp} = \prod_{i=1}^{N_v} \mathbf{e}_{\text{sp}_i}$$

Capturing Context with Global Landmarks.

Additionally, to detect the long-range dependencies, we introduce “window” nodes which act as global landmarks. Entity nodes from different windows are connected to their corresponding window node and the window nodes are connected to each other in the lexical order. Formally, for the k^{th} document, we have a set of windows $\mathcal{V}_{wn} = \{v_l^w\}_{l=1}^{N_w^k}$, where $N_w^k = \left\lceil \frac{L_k}{L_w} \right\rceil$ shows the number of windows for the k^{th} document of length L_k and L_w is the window size. For this set of windows, we have the set $\mathcal{A}_{wn} = \{\phi(v_l^w) = Y_w^n\}_{l=1}^{N_w^k}$ of window node types, s.t. $Y_w^n := \text{WINDOW}$ denotes window node type, and the set of edges to window nodes as $\mathcal{R}_{wn} = \{(\hat{y}_i^n, Y_w^e, Y_w^n) \mid \hat{y}_i^n \in \mathcal{A}_{sp}\} \cup \{(\phi(v_p^w), Y_{ww}^e, \phi(v_q^w)) \mid v_p^w, v_q^w \in \mathcal{V}_{wn}, \text{end}[v_p^w] + 1 = \text{start}[v_q^w]\}$, where $Y_w^e := \text{BELONGS-TO}$ denotes an edge from window node to its descendent entity node, $Y_{ww}^e := \text{TO}$ denotes edge type connecting the window nodes, and $\text{start}[\cdot]$ $\text{end}[\cdot]$ denote the start and end indices based on the lexical order of tokens. The feature matrix of window nodes can be written as $X_{wn} = \prod_{i=1}^{N_w^k} \rho_{[\text{CLS}]_i}$, where $\rho_{[\text{CLS}]_i}$ is the BioMedBERT embedding of the [CLS] token for the i^{th} window.

Our experimental results without employing HGT show that capturing the context between two spans probably makes significant improvements because it helps extract the short-range inter-dependencies. Therefore, we also introduce *context* nodes in the graph such that whenever there is a context between two span entities that are not farther than a distance threshold d_c , we put a context node between their respective entity nodes. Formally, we add a set of context nodes $\mathcal{A}_{ctx} = \{\phi(v_m^c) = Y_c^n\}_{m=1}^{N_c}$ s.t. $Y_c^n := \text{CONTEXT}$ denotes context node type and N_c is the number of context nodes. We make the set of context node edges as $\mathcal{R}_{ctx} = \{(\hat{y}_i^n, Y_{cb}^r, Y_c^n), (\hat{y}_j^n, Y_{ca}^r, Y_c^n) \mid i, j \in [1 \cdots N_v], \hat{y}_i^n, \hat{y}_j^n \in \mathcal{A}_{sp}, \max(\text{start}[v_i^s], \text{start}[v_j^s]) - \min(\text{end}[v_i^s], \text{end}[v_j^s]) > d_c, d_c > 1\}$, where $Y_{cb}^r := \text{BEFORE-CONTEXT}$ and $Y_{ca}^r := \text{AFTER-CONTEXT}$ are before context edge type and after context edge type respectively. For the feature matrix of context nodes we have $X_{ctx} = \big\|_{i=1}^{N_c} \mathbf{e}_{\text{ctx}(i,j)}$ s.t. $i, j \in [1 \cdots N_v]$.

At the end, for the final graph, the heterogeneous set of node types is composed as $\mathcal{A} = \mathcal{A}_{sp} \cup \mathcal{A}_{sn} \cup \mathcal{A}_{ctx}$ and the heterogeneous set of edge types is made as $\mathcal{R} = \mathcal{R}_{sp} \cup \mathcal{R}_{wn} \cup \mathcal{R}_{ctx}$. Also for the whole feature matrix we have $X = [X_{sp}; X_{wn}; X_{ctx}]$.

S4.1 HGT Module

We utilize an HGT module to encode the structural information of the document graph in its entities through message-passing (see Figure 2 in (Hu et al., 2020) for a representation of this module). This is achieved using transformers (Vaswani et al., 2017) and graph attention networks (Veličković et al., 2018) in three key components: Mutual Attention, Message Passing, and Aggregation.

Mutual Attention. Attention scores for aggregating information from node t 's neighbors are:

$$\text{Attn}(s, e, t) = \text{softmax}_{s \in \mathcal{N}_t} \left(\big\|_{i=1}^{N_h} A\text{-head}_i(s, e, t) \right) \quad (7)$$

$$A\text{-head}_i(s, e, t) =$$

$$\frac{\mu(\phi(s), \psi(e), \phi(t))}{\sqrt{d}} (K_i(s) W_a^{\psi(e)} Q_i(t)^T)$$

$$K_i(s) = K\text{-linear}_i^{\phi(s)}(h^{(l-1)}(s))$$

$$Q_i(t) = Q\text{-linear}_i^{\phi(t)}(h^{(l-1)}(t))$$

Where \mathcal{N}_t is the neighborhood of node t , d is the embedding size, $A\text{-head}_i(\cdot)$ is the i^{th} attention head, N_h is the number of heads, $W_a^{\psi(e)} \in \mathbb{R}^{\frac{d}{N_h} \times \frac{d}{N_h}}$ is the edge specific weight matrix and $K\text{-linear}_i^{\phi(s)} : \mathbb{R}^d \rightarrow \mathbb{R}^{\frac{d}{N_h}}$ and $Q\text{-linear}_i^{\phi(t)} : \mathbb{R}^d \rightarrow \mathbb{R}^{\frac{d}{N_h}}$ denote linear projections specific to the node to compute the key and query. $\mu(\phi(s), \psi(e), \phi(t)) \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{R}| \times |\mathcal{A}|}$ is the significance factor for each meta-relation. These node- and edge-specific projections and attention matrices help with optimal modeling of distribution differences among different types of nodes and edges for information aggregation.

Message Passing & Aggregation. HGT allows edge-type-specific message passing for each node:

$$\text{Msg}(s, e, t) = \big\|_{i=1}^{N_h} M\text{-head}_i(s, e, t) \quad (8)$$

$$M\text{-head}_i(s, e, t) = M\text{-linear}_i^{\phi(s)}(h^{(l-1)}(s)) W_m^{\psi(e)}$$

Where, $W_m^{\psi(e)} \in \mathbb{R}^{\frac{d}{N_h} \times \frac{d}{N_h}}$ are edge-specific weights and $M\text{-linear}_i^{\phi(s)} : \mathbb{R}^d \rightarrow \mathbb{R}^{\frac{d}{N_h}}$ is node-specific linear projection. Finally, messages are aggregated across all source nodes within the neighborhood of a target node t to update it as:

$$h^l(t) = G\text{-head}^{\phi(t)}(\sigma(\tilde{h}^l(t))) + h^{(l-1)}(t) \quad (9)$$

$$\tilde{h}^l(t) = \sum_{s \in \mathcal{N}_t} \text{Attn}(s, e, t) \cdot \text{Msg}(s, e, t) \quad (10)$$

Where σ is a non-linear function and $G\text{-head}^{\phi(t)} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ a linear projection for t .

S5 Implementation Details

All the experiments have been performed on a single NVIDIA-A100 GPU. For maximum span length, we choose a threshold of 7 tokens that falls within the 97th percentile of entity length distribution in the training split. We also experimented with the threshold of 20 which covers 99.99 percentile. Since entities with very long token lengths are rare, we find that 7 gives better performance and is much faster to train. As our base encoder, we use the BioMedBERT variant microsoft/BioMedNLP-BioMedBERT-base-uncased-abstract-fulltext.

To obtain span pair representation, we also experiment with concatenating max pooled or mean pooled or attention-based pooling of the embeddings of all the tokens constituting a span. Further, for span pair representation, we experiment with

element-wise sum and difference to capture the interaction between entity spans. For incorporating additional context, we also use left-right pooling, where we max-pooled all the tokens within a fixed-sized window to the left and right of each span (with window sizes 10, 100). However, these experiments have lower performance.

We use the Adam Optimizer with a linear warmup and learning rate of 0.0008 and a dropout of 0.35 (hyperparameter range 0.2-0.35) after the entity and relation classifier. The hyperparameters were manually selected based on some previous works and some experimentation. The weights of the entity and relation decoder are initialized using default uniform distribution. We restrict span-width to a max of 7 with the dimension of the span-width embedding being 7. We also experimented with 1–2 linear layers with ReLU activation for obtaining span representation and 1–2 layers for mention and relation decoder. We use 1000 hidden units for each linear layer (we also experimented with 100–800 units and found the best performance with 1000 units. We are not able to increase these further due to GPU-memory constraints). We use a context window of length $L_w = 512$, this is also the maximum token length the BioMedBERT model can process.

For GNN-related hyperparameters, we use two attention heads (we experimented with 2–8 heads) and 2 layers (hyperparameter range 1–3). We run 2 iterations of GNN-based refinement (hyperparameter range = 1–2) and use a dropout of 0.3 after each HGT layer. For selecting the most confident edges from SPANTREX, we use a threshold τ of 0.4 for the prediction probability (hyperparameter range = 0.1–0.5). For creating dummy nodes, we use a pooling distance $d_c = \delta \cdot \text{document-length}$, where $\delta=0.5$ (hyperparameter range 0.5–1). Finally, we use a residual coefficient of 1.0 with the span embeddings to combine them with GNN-refined node embeddings and obtain final entity embeddings (hyperparameter range for residual coefficient = 0.5–1.0).

We train the model for 20 epochs (we experimented with up to 50 epochs, the models don’t show improvement after epoch 20) and save the best-performing model on the validation split after each epoch. We use a batch size of 8 (and experiment with batch sizes in the range 1–16, where 16 is quite large given the GPU memory limitations and quadratic complexity of relation extraction). We allow the base transformer BioMedBERT to

fine-tune its weight using a learning rate of $3e-05$. We only use the entity loss \mathcal{L}_n for the first two epochs and after that use the combined loss $\mathcal{L} = \mathcal{L}_n + \mathcal{L}_r$ since the model does not generate any entities for the first few epochs.

Model	Time		#Parameters	Disk Space (GB)
	Train (hrs)	Inference (sec)		
SPANTREX	7.3	14	1.2M	0.44
GRAPHTREX	10.8	14	1B	4.35

Table S5: Additional Computational Metrics for models trained on the I2B2 corpus

Our method GRAPHTREX takes approximately 11 hours for training and on average 14 seconds for inference on each document, and around 30 minutes on the full test split comprising 120 documents. The computational metrics are summarized in Table S5.

For the multi-head attention baseline, we use 128 heads of 64 dimensions and find that the performance decreases by increasing/decreasing the heads.

S6 Additional Robustness Checks

S6.1 Performance Across Additional Encoders and BioMedBERT-UMLS

Table S6 reports performance across various encoders pre-trained on clinical texts on the I2B2 corpus. These include MimicEntityBERT (Lin et al., 2021), a fine-tuned variant of BioMedBERT, that uses an entity-masking strategy to mask events and time expressions and achieves SOTA results on the THYME corpus. However, neither this nor other tested models outperform BioMedBERT on NER for the I2B2 2012 dataset.

We also introduce BioMedBERT-UMLS, a span-based NER model with external knowledge infusion. For this, we integrate concepts extracted from UMLS (Bodenreider, 2004) using MetaMap (Aronson et al., 1994). We then construct an induced subgraph of these concepts based on the UMLS semantic network, following Lai et al. (2021). Nodes (concepts) are initialized using pre-trained knowledge-infused embeddings from Maldonado et al. (2019). We model this graph using a Relational Graph Convolutional Network (RGCN) with two-hop message passing. The final node embeddings are concatenated with BioMedBERT embeddings based on span overlap and classified using a two-layer fully connected decoder. This approach

sets a new state-of-the-art in timex extraction while achieving comparable performance on event extraction.

Model	Event (F1)	TimEx (F1)
BioClinicalBERT (Alsentzer et al., 2019)	86.15	88.62
BlueBERT (Peng et al., 2019)	86.51	88.63
EntityBERT (Lin et al., 2021)	90.18	89.50
BioMedBERT	90.57	90.62
BioMedBERT (UMLS-fusion)	90.05	91.63

Table S6: Comparing encoders pre-trained on clinical text on the NER task of the I2B2 dataset with SPANTREX.

We also present more details of additional encoders presented in Table 5 in the expanded Table S7, allowing comparisons across span identification (EI) and entity typing (EC).

S6.2 Pipeline Approach.

We experiment with a pipeline approach that first extracts entities using BioMedBERT-UMLS, a state-of-the-art model for timex extraction, described in the last paragraph. These extracted entities are then passed to an independent relation extraction module, where we apply the method of Zhong and Chen (2021), which achieves state-of-the-art performance on I2B2 2012 relation classification (Cheng and Weiss, 2023).

Unlike end-to-end relation extraction, relation classification is simpler as it only classifies entity pairs that are known to participate in a relation and, therefore, excludes the majority No-relation class representing entity pairs without a clear temporal link. For relation extraction, we add this class by creating multiple negative training samples for each entity pair. Each training sample is constructed by inserting special marker tokens around entities for better-aligning entity boundary representations. While this approach is effective for sentence-level relation extraction, it becomes computationally prohibitive at the document level as it requires generating marker-augmented sentence pairs for every possible entity pair. Therefore, we restrict our experiment to same-sentence relation extraction, focusing on same-sentence entity pairs. Our comparison reveals that this model severely under-performs when tasked with the additional complexity of the No-relation class, with an F_1 score of 10.79%. In comparison, the F_1 score of GRAPHTREX on sentence-level relations

is 59.67%.

S6.3 Performance on E-E relations.

As noted by Han et al. (2020), predicting E-E TLinks or the temporal relations between two events is more challenging than predicting relations when one of the participating pairs is a timex. This is because the event spans do not explicitly specify the starting or the ending time of an event. We hypothesize that the heterogeneous modeling over various event types helps the model learn the domain-specific constraints. We find that GRAPHTREX indeed improves the performance of E-E relations on the I2B2 corpus with the temporal F_1 score over the E-E relations being 49.85%, while for SPANTREX, the score is 46.27%. Here, we evaluate these two models over all E-E pairs and not just on the nearby pairs as in Han et al. (2020).

S6.4 Impact of Node Types across E3C

We also assess the impact of different nodes in HGT construction on the performance across the E3C corpus in Table S8 and find that Window nodes are not helpful for this corpus. This is expected since, on average, a document in E3C spans only a single window (the maximum window length is 512 tokens, and the average document length in the test set is 392 tokens). Even the longest document in the test set spans only two windows. When we exclude the context nodes but retain the window nodes, these nodes still connect the entities, allowing information propagation. Including both leads to a slight drop in comparison, and excluding both leads to a further marginal drop.

S7 Qualitative Analysis

We perform a qualitative analysis by manually inspecting the output of our GRAPHTREX model. We present an example case study in Figure S3 representing one of the documents. To comply with data usage agreements, the original text is not shared, and some node captions are modified while preserving the original meaning. For visualization clarity, we invert the *After* relations to *Before* by flipping the participating entities in the pair. We also remove duplicates after inverting such relations. Note that GRAPHTREX produces more balanced relations compared to SPANTREX which extracts more *Overlap* pairs compared to *Before/After*. Interestingly, there is an error in the ground truth annotations—the entity ‘discharged’ Overlaps with the admission date. This creates

Model	EVENT		TimEx		TLink		
	EI	EC	EI	EC	RE		
	F_1	Acc	F_1	Acc	P	R	F_1
GRAPHTREX	89.55	80.99	90.06	81.21	78.21	61.42	68.81
BERT-SPANTREX	85.87	71.10	85.79	70.93	75.94	48.63	59.30
BERT-GRAPHTREX	85.83	71.86	86.25	72.64	77.45	52.30	62.44
RoBERTA-SPANTREX	88.77	80.34	90.81	82.58	79.98	52.05	63.06
RoBERTa-GRAPHTREX	86.42	75.30	89.62	80.05	78.22	54.88	64.50
Clinical-Longformer	86.62	77.24	91.03	81.76	82.39	52.10	63.84

Table S7: Expanded Table for Robustness checks with additional encoders for the I2B2 corpus.

Model	EVENT	TLink
	F_1	F_1
SPERT	78.85	13.63
SPANTREX	81.30	22.55
GRAPHTREX		
Excluding Window Nodes	82.1	23.48
Excluding Context Nodes	81.73	23.39
Excluding Both	78.38	22.73
Including Both Nodes	81.46	22.97

Table S8: The effectiveness of Window and Context nodes on the E3C corpus.

an inconsistency by transitivity, where the dates ‘10/5/95’ and ‘10/9/95’ are overlapping. This issue is also present in SPANTREX predictions. However, GRAPHTREX addresses the issue. For this example, all the graphs demonstrate structural consistency (no cycles involving directed arrows). SPANTREX misses two important wellness indicators—‘tolerated’ and ‘treatment’. Both models also exclude the entity ‘well’, a property of ‘tolerated’ in the text. They also separate ‘daily for four days’ into ‘daily’ and ‘four days’ and connect them with *Overlap* relation, thereby still preserving the meaning.

We also find that some ‘errors’ by standard metrics reflect the strengths of our model, revealing gaps in the gold standard. For example, given the co-referring pairs such as (‘discharge’, ‘discharged’), (‘admission’, ‘hospitalization’), our model can correctly identify the second mention as a relevant entity type even though only the first span has been labeled in the ground truth. This yields additional correct relations with the missing entities that are also missing in the gold standard. Lastly, we note that for some of the documents, the

model inconsistently predicts the relation between discharge and admission as *Before*. Approaches to infuse real-world commonsense knowledge to improve this might be an interesting future direction.

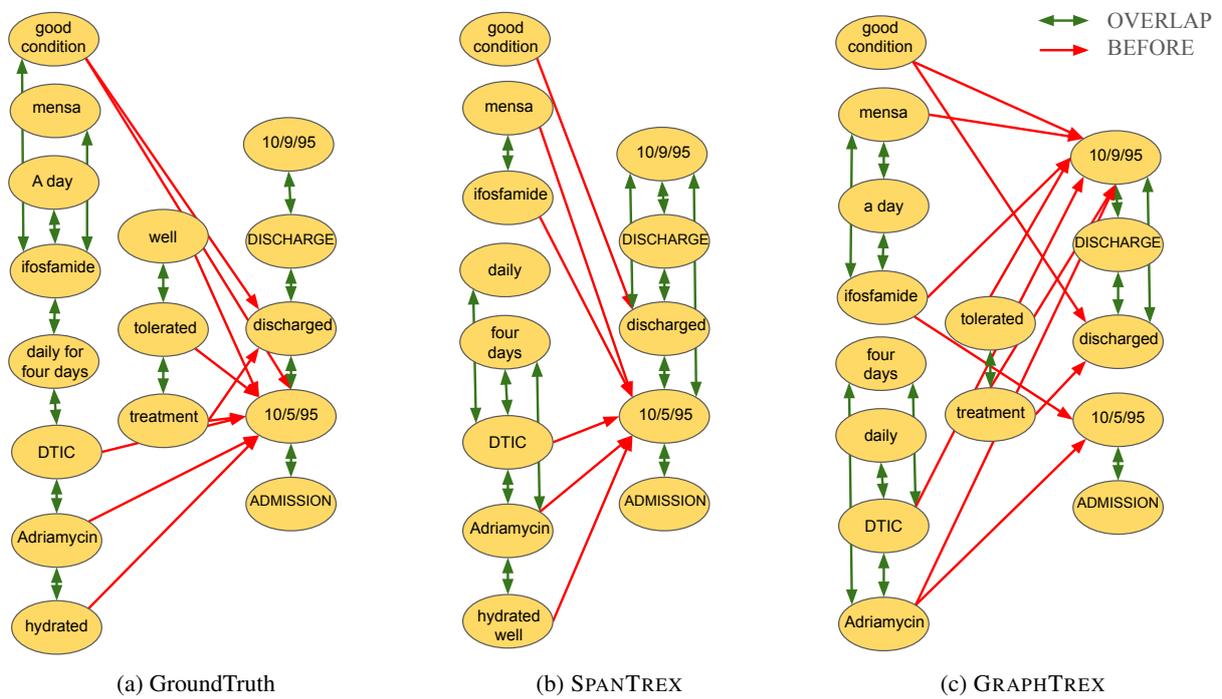


Figure S3: Ground Truth vs. System Predictions for Temporal Graphs. The figure compares the reference and predicted temporal graphs for a clinical document from the I2B2 corpus.