Textual Entailment for Temporal Dependency Graph Parsing

Jiarui Yao[1] **Steven Bethard**[2] **Kristin Wright-Bettner**[3]
Eli Goldner[1] **David Harris**[1] **Guergana Savova**[1]

[1] Boston Children's Hospital and Harvard Medical School
[2] University of Arizona [3] University of Colorado at Boulder
{firstname.lastname}@childrens.harvard.edu
bethard@email.arizona.edu kristin.wrightbettner@colorado.edu

Abstract

We explore temporal dependency graph (TDG) parsing in the clinical domain. We leverage existing annotations on the THYME dataset to semi-automatically construct a TDG corpus. Then we propose a new natural language inference (NLI) approach to TDG parsing, and evaluate it both on general domain TDGs from wikinews and the newly constructed clinical TDG corpus. We achieve competitive performance on general domain TDGs with a much simpler model than prior work. On the clinical TDGs, our method establishes the first result of TDG parsing on clinical data with 0.79/0.88 micro/macro F1. Our code is available at https://github.com/Jryao/thyme_tdg.

1 Introduction and Background

Temporal information extraction from text is an important part of natural language understanding. Many works have framed temporal relation extraction (RE) as the task of identifying temporal relations between pairs of events, or an event and a time expression (TIMEX3) (Pustejovsky et al., 2003a,b; Cassidy et al., 2014; Styler IV et al., 2014; Ning et al., 2018; Ballesteros et al., 2020; Lin et al., 2021). This pairwise framing can make it hard to decide when to annotate a temporal relation, and the resulting timelines are usually fragmented (Kolomiyets et al., 2012) as not all events or TIMEX3s are linked to each other. Heuristics are typically applied to constrain the search space of pairwise relations, both for annotators and machine learning models. For example, many annotation efforts have constrained temporal relations to adjacent sentences: TempEval (Verhagen et al., 2007, 2010; UzZaman et al., 2013), Clinical TempEval, (Bethard et al., 2015, 2016, 2017), and TimeBank-Dense (Cassidy et al., 2014).

A more principled approach is to model the temporal information in a document as a dependency tree structure (Kolomiyets et al., 2012; Zhang and Xue, 2018b). This approach was extended by Yao et al. (2020) to *temporal dependency graph* (TDG) structure for a more comprehensive representation. An example is shown in Figure 1. With this approach, human annotators inspect each entity and find at most 2 reference times. A complete TDG can be constructed from these decisions. The automatic temporal RE task then becomes a parsing task: produce a TDG as output given a text as input. TDG datasets have been constructed for news articles (Yao et al., 2020) and contracts (Mathur et al., 2022). The current state-of-the-art (SoTA) TDG model (Mathur et al., 2022) reports 0.77 F1 score in the general domain, and 0.64 F1 on the contract dataset, showing the learnability of the TDG approach in those two domains.

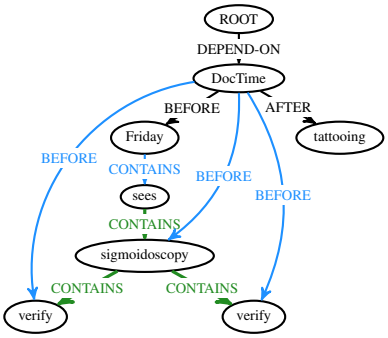

Figure 1: TDG representation for "We will have Dr. Lee perform a flexible **sigmoidoscopy** to **verify** the **tattooing** and to **verify** the location when he **sees** her on Friday." DocTime is the Document Creation Time.

In the current work, we make the following contributions:

- We bring TDGs to the clinical domain, by converting the pairwise annotations over the Mayo Clinic electronic health record (EHR) notes in the widely used THYME corpus (Styler IV et al., 2014) to TDGs.[1]

[1] Our THYME-TDG dataset will be available to the research community under the THYME data use agreement procedure.

Proceedings of the 5th Clinical Natural Language Processing Workshop, pages 191–199

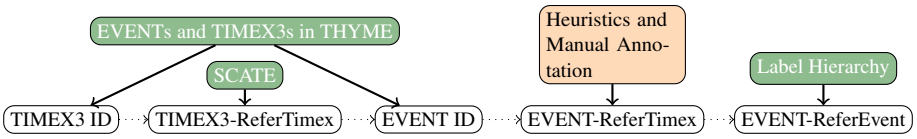

Figure 2: Overview of converting THYME pairwise annotations to TDGs. ID refers to identification, ReferTimex and ReferEvent denote reference timex and reference event respectively. Only the step of identifying reference timex for events requires manual annotation.

- We develop an natural language inference (NLI)-based TDG parser that is much simpler than prior TDG parsers, yet achieves performance competitive with the state-of-the-art in the general domain. On the newly constructed clinical TDG dataset, this parser also achieves strong performance.

Our TDG parser is inspired by works applying NLI-based methods to other information extraction tasks, including relation extraction (Sainz et al., 2021) on the TACRED data set (Zhang et al., 2017), event argument extraction (Sainz et al., 2022) on the ACE (Walker et al., 2006) and WikiEvents (Li et al., 2021) datasets, and biomedical relation extraction (Xu et al., 2022).

2 Creating a Clinical TDG Corpus

A temporal dependency graph (TDG) is defined as a 4-tuple (T, E, M, L), where T is a set of TIMEX3s, E is a set of EVENTs, M is a set of pre-defined "meta" nodes (e.g. ROOT), and L is a set of edges. The definitions and guidelines[2] of Yao et al. (2020) describe the steps to create a TDG from EVENTs and TIMEX3s: (1) For each TIMEX3 t, if t is locatable (i.e., t is not a QUANTIFIER, DURATION or SET), find its reference time expression (*reference timex*), otherwise assign no reference timex; (2) For each EVENT e, find its *reference timex*; and (3) For each EVENT e, find its *reference event* if there is one. Fig. 1 shows examples of such reference decisions, where the reference timex of "Friday" is "DocTime", the reference timex of "sigmoidoscopy" is "DocTime", and the reference event of "sigmoidoscopy" is "sees".

We semi-automate this TDG construction process by leveraging the existing annotations over the THYME corpus (Styler IV et al., 2014). Our approach is visualized in Fig 2. First, we take all the EVENTs and TIMEX3s in the THYME corpus as the building blocks of the graph. In the following steps, we include as many TLINKs (temporal links) from the THYME corpus as possible to maintain the richness and informativeness of the THYME annotations. In some cases, we reverse the TLINK label (e.g. <e_1 BEFORE e_2> becomes <e_2 AFTER e_1>) to make the final graph structure simpler and the annotation process easier (see Appendix A.1).

Temporal Operator	TLINK Labels
Last	BEFORE
Next	AFTER
Before	BEFORE
This	OVERLAP
After	AFTER

Table 1: Mapping SCATE temporal operators to THYME temporal relations.

Identifying the Reference Timex for a TIMEX3. TLINKs between two TIMEX3s are not annotated in the THYME corpus as the temporal relations between a pair of TIMEX3s can be inferred if their normalized values are available.[3] For a locatable TIMEX3, we use the gold temporal operators annotations from the Semantically Compositional Annotation of Time Expressions (SCATE; Bethard and Parker, 2016) to get a TIMEX3-TIMEX3 relation by mapping temporal operators to temporal relations as shown in Table 1.

Identifying the Reference Timex for an Event. Given an event e, we choose the reference timex of e among the TIMEX3s linked to e in the original THYME corpus via TLINKs. If there is only one TIMEX3 temporally related to e, that TIMEX3 is automatically assigned as the reference timex of e. If there are multiple TIMEX3 temporally related to e, but only one TIMEX3 CONTAINS e, that TIMEX3 is automatically selected as the reference timex of e. Otherwise, the instance's reference timex is manually annotated. If e is not TLINKed to any TIMEX3s, DocTime is selected

[2] https://github.com/Jryao/temporal_dependency_graphs_crowdsourcing

[3] https://clear.colorado.edu/compsem/documents/THYME_guidelines.pdf

	Train	Dev	Test
	THYME-TDG		
TDG	1,822	927	998
TIMEX3-TIMEX3	4,328	2,286	2,216
EVENT-TIMEX3	37,042	20,100	18,499
EVENT-EVENT	10,073	5,798	5,443
EVENT-EVENT (THYME+)	11,558	6,579	6,225
EVENT-EVENT %	87.2%	88.1%	87.4%
	general-TDG		
TDG	400	50	50
TIMEX3-TIMEX3	1,952	325	209
EVENT-TIMEX3	12,047	1,717	1,015
EVENT-EVENT	8,725	1,298	706

Table 2: THYME-TDG and general-TDG distributions. EVENT-EVENT % is the percent of EVENT-EVENT relations in THYME+ represented in THYME-TDG.

as its reference timex with the relation between an event and DocTime (DocTimeRel) as the label.

Identifying the Reference Event for an Event. To find the reference event for e among the EVENTs that are TLINKed to e, we define a label hierarchy[4] based on the specificity of different temporal relations to facilitate consistent annotation:

BEGINS-ON, ENDS-ON >
CONTAINS, CONTAINS-SUBEVENT >
BEFORE >
OVERLAP, NOTED-ON

If the candidates have the same level of specificity, we choose the one closest to e in textual order.

3 Corpus Statistics

Following the Clinical TempEval tasks, we use the colon cancer set of the THYME+ corpus and the same data splits as described in Wright-Bettner et al. (2020). Our statistics on the Development (Dev) set show that there are on average 155 events per note. Since 91.5% of the TLINKs occur within-section, and all the cross-section TLINKs are CONTAINS-SUBEVENT relations, we split each note into sections, drop the cross-section TLINKs, and create one TDG per section.

With our carefully designed conversion method, we automatically translated most of the pairwise annotations in the THYME+ corpus into TDGs. There were only 72, 35 and 40 events in the Train, Dev and Test sets, respectively, which could not be automatically assigned reference times. Because of this small number, one domain expert manually annotated these missing TDG edges. Statistics of the final THYME-TDG corpus are in Table 2. The EVENT-EVENT rows show that more than 87% of the original event-event TLINKs are represented in the THYME-TDG corpus. Table 2 also shows the statistics of the publicly available TDG dataset (general-TDG; Yao et al., 2020).

4 TDG Parsing

The task of temporal dependency graph parsing is to find the *parent node(s)* for each *child node*, where the child node can be an EVENT or a TIMEX3. We cast the TDG parsing task as a textual-entailment (NLI) task. Given a child node x_i and a list of candidate parent nodes $\{y_1, y_2, \ldots, y_i, \ldots, y_n\}$, our model first verbalizes the possible relations between $<x_i, y_i>$ to generate *hypotheses* using a list of pre-defined templates (shown in Appendix A.3). For example, the verbalization for the BEFORE relation is: x_i *happened before* y_i. To get the *premise*, we concatenate the sentence that contains the child node and the one that contains the parent node.[5] If y_i is not the parent of x_i, the relation label between x_i and y_i will be NO-EDGE. We also add descriptions after the premise regarding the distance between two nodes, following previous works (Zhang and Xue, 2018a; Ross et al., 2020; Yao et al., 2020).[6] Then, we run the NLI model to obtain the probability the premise entails/contradicts/is neutral to the hypothesis.

In the **training** stage, we finetune a pre-trained NLI model on the entailment data generated from the TDG training data. In a TDG, each event can have at most two parents: a reference timex, and a reference event. That is, most of the time, a candidate parent node is not the gold parent for the child, which means the relation between a child node x_i and a candidate parent y_i is NO-EDGE in most cases. To obtain a relatively balanced training set, we first divide the candidate parent nodes into two sets: A and B, where A contains the gold parent nodes for the current child node, and B contains the rest. Then, we randomly sample N_E examples

[4]While the rest of the hierarchy reflects actual specificity, we do not claim that CONTAINS is more specific than BEFORE. We gave preference to CONTAINS here to capitalize on THYME's narrative containers (Styler IV et al., 2014). This is also why we chose CONTAINS as the preferred TLINK type for selecting a reference timex.

[5]Only one sentence will be needed if the two nodes are in the same sentence.

[6]Our preliminary experiments show the additional features can increase the model performance by about 2%.

from A, and N_C examples from B. For each instance sampled from A, we generate an *entailment* example using the gold label, and randomly sample N_C incorrect labels to generate *neutral* examples. For each instance sampled from B, we generate one *contradiction* example (see Appendix A.3 for an example). In the **inference** stage, for each candidate parent node y_i of a child node x_i, we verbalize all possible relations between them and pick the candidate parent with the highest entailment probability as the final parent for x_i.

5 Experimental Setup

We evaluate our model on the two TDG data sets: the general-TDG and THYME-TDG (clinical-TDG).[7] When sampling the training data, we set N_E to 1 and N_C to 3. For general-TDG, to generate the reference timex candidates for each TIMEX3 or EVENT, we include all the TIMEX3s in the document; to generate reference event candidates for each EVENT, we include all the EVENTs from the beginning of the document to two sentences after the child node. For clinical-TDG, we include candidates in the window of 6 sentences before and 4 sentences after the child node. For both data sets, our candidate parent window setting covered > 99% of the cases.

For the general domain TDG parsing, we finetune the roberta-large-mnli (Liu et al., 2019; Williams et al., 2018) model via HuggingFace (Wolf et al., 2020) for 3 epochs. For the clinical domain TDG parsing, we finetune the PubMedBERT-mnli-snli-scinli-scitail-mednli-stsb (Deka et al., 2022) model for 3 epochs. For model initialization, we experimented with 5 random seeds: {42, 52, 62, 72, 82}. See Appendix A.2 for other hyperparameters.

We use gold EVENTs and TIMEX3s as input for the TDG parsers. Parsed <*child, relation, parent*> triples are compared against gold triples to compute F1 scores. On the general-TDG, we report the average F1 scores across all documents (macro-F) following previous practice (Zhang and Xue, 2018a; Yao et al., 2020; Ross et al., 2020; Mathur et al., 2022). On the clinical-TDG data set, we report both macro- and micro- F1 scores.

[7]The TDG dataset of 100 contracts used in Mathur et al. (2022) is not publicly available to the best of our knowledge.

	Dev	Test
BERT-Ranking (Ross et al., 2020)*	0.62	0.71
DocTime (Mathur et al., 2022)*	**0.69**	**0.77**
NLI-based TDG (best)	<u>0.67</u>	<u>0.75</u>
NLI-based TDG (average)	0.66	0.74

Table 3: TDG parsing F1 scores on the general-TDG. Best results bolded, second best underlined. * indicates results from (Mathur et al., 2022). "Best" and "average" refer to the best and average results across 5 seeds.

	Dev	Test
NLI-based TDG (best)	0.88 (0.79)	0.88 (0.79)
NLI-based TDG (average)	0.87 (0.79)	0.88 (0.79)

Table 4: TDG parsing macro F1 (micro F1) scores on the clinical-TDG data set.

6 Results and Discussion

Table 3 shows our general-TDG results. We compare our NLI-based TDG model with two existing supervised models trained for the TDG parsing task: the BERT-Ranking model (Ross et al., 2020)[8] and the DocTime model (Mathur et al., 2022). Both our NLI-based TDG (best) and (average) models outperform the BERT-Ranking model by a large margin, suggesting the advantages of our NLI approach. The NLI-based TDG (average) model and BERT-Ranking model report the average scores of 5 runs, however it is unclear whether the DocTime results are the best or the average.

Compared to the DocTime model (Mathur et al., 2022), our NLI-based TDG model (best) achieved slightly lower but competitive performance, while being much simpler. The DocTime model contains 3 graph neural networks and relies on off-the-shelf NLP tools including a co-reference resolution model, a dependency parser, a pre-trained model for sentence embeddings, and a document-level Rhetorical Structure Theory parser. Mathur et al. (2022) did not list exact tools or configurations (e.g., what is the model used for coreference resolution?) and the code is not publicly available, so it's very hard to re-implement or apply this model to other data sets currently.

We evaluated our NLI-based TDG approach on the newly created clinical-TDG data set (Table 4). This is the first result with a graph algorithm on a clinical temporal relation dataset. Thus, our re-

[8]This model was not evaluated on the TDG data set by the authors, Mathur et al. (2022) ran the experiments on general-TDG and reported the results in their publication.

sults serve as a baseline for future research. Our NLI-based parser achieved promising results on the clinical-TDG data, showing both the utility of this dataset, and the generalizability of our TDG parser.

7 Conclusion

We explore TDG representation and parsing in the clinical domain. We convert the pairwise annotations over the Mayo Clinic EHR notes in the THYME corpus to TDGs semi-automatically. We then develop a NLI-based TDG parser that is much simpler than prior TDG parsers, yet achieves performance competitive with the SoTA in the general domain. On the clinical TDG data set, our parser also achieves strong performance, which can serve as a baseline for future research on clinical TDG parsing.

Limitations

We finetuned pre-trained NLI models for TDG parsing. Both data sets we used were in English. To apply this model to other languages and to get the best results, pre-trained NLI models or NLI data sets might be required for the new language. Templates to verbalize the temporal relations in the new language are also required.

The clinical data set (i.e. THYME) we used in this work only contains EHRs from one institution: Mayo Clinic. Clinicians from different hospitals can have different writing style or use different templates when writing the notes. Future work should test the TDG representation and parsers on EHRs from other institutions, and EHRs of different patient populations.

Acknowledgments

The research was supported by NIH (R01LM010090, R01LM01348). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Library Of Medicine or the National Institutes of Health.

References

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

Steven Bethard, Leon Derczynski, Guergana Savova, James Pustejovsky, and Marc Verhagen. 2015. SemEval-2015 task 6: Clinical TempEval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 806–814, Denver, Colorado. Association for Computational Linguistics.

Steven Bethard and Jonathan Parker. 2016. A semantically compositional annotation scheme for time normalization. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3779–3786, Portorož, Slovenia. European Language Resources Association (ELRA).

Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. SemEval-2016 task 12: Clinical TempEval. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1052–1062, San Diego, California. Association for Computational Linguistics.

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.

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 501–506, Baltimore, Maryland. Association for Computational Linguistics.

Pritam Deka, Anna Jurek-Loughrey, et al. 2022. Evidence extraction to validate medical claims in fake news detection. In *International Conference on Health Information Science*, pages 3–15. Springer.

Oleksandr Kolomiyets, Steven Bethard, and Marie-Francine Moens. 2012. Extracting narrative timelines as temporal dependency structures. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 88–97, Jeju Island, Korea. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

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.

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

Puneet Mathur, Vlad Morariu, Verena Kaynig-Fittkau, Jiuxiang Gu, Franck Dernoncourt, Quan Tran, Ani Nenkova, Dinesh Manocha, and Rajiv Jain. 2022. DocTime: A document-level temporal dependency graph parser. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 993–1009, Seattle, United States. Association for Computational Linguistics.

Qiang Ning, Hao Wu, and Dan Roth. 2018. A multi-axis annotation scheme for event temporal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1318–1328, Melbourne, Australia. Association for Computational Linguistics.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003a. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

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

Hayley Ross, Jonathon Cai, and Bonan Min. 2020. Exploring Contextualized Neural Language Models for Temporal Dependency Parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8548–8553, Online. Association for Computational Linguistics.

Oscar Sainz, Itziar Gonzalez-Dios, Oier Lopez de Lacalle, Bonan Min, and Eneko Agirre. 2022. Textual entailment for event argument extraction: Zero- and few-shot with multi-source learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2439–2455, Seattle, United States. Association for Computational Linguistics.

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero and few-shot relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1199–1212, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

William F. Styler IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C de Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, and James Pustejovsky. 2014. Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143–154.

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

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 task 15: TempEval temporal relation identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80, Prague, Czech Republic. Association for Computational Linguistics.

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

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

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Kristin Wright-Bettner, Chen Lin, Timothy Miller, Steven Bethard, Dmitriy Dligach, Martha Palmer, James H. Martin, and Guergana Savova. 2020. Defining and learning refined temporal relations in the

clinical narrative. In *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, pages 104–114, Online. Association for Computational Linguistics.

Jiashu Xu, Mingyu Derek Ma, and Muhao Chen. 2022. Can nli provide proper indirect supervision for low-resource biomedical relation extraction?

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

Yuchen Zhang and Nianwen Xue. 2018a. Neural ranking models for temporal dependency structure parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3339–3349, Brussels, Belgium. Association for Computational Linguistics.

Yuchen Zhang and Nianwen Xue. 2018b. Structured interpretation of temporal relations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45.

A Appendix

A.1 Annotation Details

Given a TLINK <event A, r, event B> in the THYME+ corpus, where event A is the Source (parent), event B is Target (child), and r is the label between them, we reverse the TLINK label in the following case: event B is the Target in multiple TLINKs, while event A is only TLINKed with event B.

In example 1 below, instead of looking for the reference event of **colonoscopy** between **mass** and **bleeding**, it's easier to make **colonoscopy** the reference event of **mass** and **bleeding** by reversing those two TLINKs: <mass, NOTED-ON, colonoscopy> becomes <colonoscopy, NOTED-ON-INV, mass>, and <bleeding, NOTED-ON, colonoscopy> becomes <colonoscopy, NOTED-ON-INV, bleeding>, with "INV" indicating "inverse". The subgraph representation for those 3 TLINKs is showed in Figure 3.

In example 2 below, it's not clear which event among **report, pathology, values** and **notes** should be the reference event of **review** as they all have the same temporal relation with **review**. However, if we reverse those TLINKs, then **review** will become the reference event of the 4 other events, as shown in Figure 4.

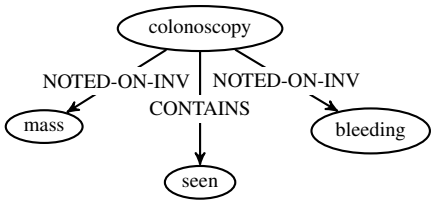

Figure 3: Final graph representation of Example 1.

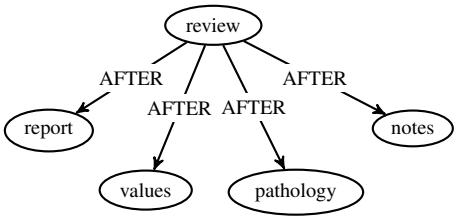

Figure 4: Final graph representation of Example 2.

1. *Review of the **colonoscopy** reports indicates that approximately 10-cm from the anal verge a 3- to 4-cm **mass** was **seen** with **bleeding**.*
   - <mass, NOTED-ON, colonoscopy>
   - <colonoscopy, CONTAINS, seen>
   - <bleeding, NOTED-ON, colonoscopy>

2. *I have had the opportunity to **review** the operative **report**, surgical **pathology**, laboratory **values**, and **notes**.*
   - <report, BEFORE, review>
   - <pathology, BEFORE, review>
   - <values, BEFORE, review>
   - <notes, BEFORE, review>

A.2 Implementation Details

For the general-TDG data set, we carried out a grid-search of training epochs in {**3**, 4, 10, 20}, batch size in {16, **32**, 64}, maximum sequence length in {64, **128**}, learning rate in {1e-5, 2e-5, **3e-5**}, and weight decay in {0.1, 0.2, **0.3**}, final parameter settings are in bold.

For the clinical-TDG data set, we experimented with training epochs in {**3**, 4}, batch size in {**16**, 32}, learning rate in {**1e-5**, 2e-5, 3e-5}, and weight decay in {**0.1**, 0.2, 0.3}, final parameter settings are in bold.

Experiments were run on an NVIDIA Titan RTX GPU cluster of 7 nodes. It took 80 - 90 minutes to run one training epoch for both data sets.

A.3 Templates

The templates we used to verbalize temporal relations are listed in Table 5 and Table 6.

We give a concrete example to show how we generate the NLI instances from our TDG data sets. Given an event e_i, let $\{e_1, e_2, e_3\}$ be its candidate reference events, e_2 be the gold reference event, and let BEFORE be the gold relation between e_2 and e_i. The following are the NLI instances we can generate for this example:

- *Entailment*: e_2 happened before e_i.
- *Neutral*: e_2 happened at around the same time as e_i.
- *Contradiction*: During e_3, e_i happened.

Both *Entailment* and *Neutral* examples are generated with the gold candidate event e_2, the difference is that the *Neutral* instance has the wrong label that is randomly sampled from the label set. The *Contradiction* example is generated by randomly sampling an incorrect reference event from the candidates with a random label.

Please note that in both the training and inference stage, entity type constraints are applied when verbalizing a temporal relation. For example, in the general-TDG data set, "included" is only used for event-timex pairs. Therefore, when verbalizing event-event relations, the "included" label will be ignored.

A.4 Features

The linguistic features we used are showed in Table 7.

Label	Template
before	{subj} happened before {obj}
after	{subj} happened after {obj}
overlap	{subj} happened at around the same time as {obj}
included	{subj} happened {obj}
Depend-on	{subj} depended on {obj}

Table 5: Templates we used to verbalize temporal relations in the general-TDG data set. {subj} and {obj} are placeholders for entities.

Label	Template
BEFORE	{subj} happened before {obj}
AFTER	{subj} happened after {obj}
OVERLAP	{subj} happened at around the same time as {obj}
CONTAINS-SUBEVENT	{obj} is a sub-event of {subj}
CONTAINS-SUBEVENT-INV	{subj} is a sub-event of {obj}
NOTED-ON	The {obj} test showed the result {subj}
NOTED-ON-INV	The {subj} test showed the result {obj}
AFTER/OVERLAP	{subj} happened after or overlap {obj}
CONTAINS	During {subj}, {obj} happened
CONTAINS-INV	During {obj}, {subj} happened
Depend-on	{obj} depended on {subj}
BEGINS-ON	{subj} begins on {obj}
ENDS-ON	{subj} ends on {obj}

Table 6: Templates we used to verbalize temporal relations in the clinical-TDG data set. {subj} and {obj} are placeholders for entities. "INV" means "inverse", for example, CONTAINS-INV is the inverse of CONTAINS.

Description
Same sentence
Parent sentence before child sentence
Parent sentence after child sentence
No reference event
Parent is Root
Parent is DCT
Parent is the immediately previous node of the child node
Parent is two nodes before the child node in textual order
Parent is the immediately succeeding node of the child node
Parent node after the child node in text order

Table 7: We describe the sentence distance and node distance between two nodes in natural language, as listed in this table.