EventFull: Complete and Consistent Event Relation Annotation

Alon Eirew **Eviatar Nachshoni** **Aviv Slobodkin** **Ido Dagan**
Bar-Ilan University
{alon.eirew, eviatarn, lovodkin93}@gmail.com
dagan@cs.biu.ac.il

Abstract

Event relation detection is a fundamental NLP task, leveraged in many downstream applications, whose modeling requires datasets annotated with event relations of various types. However, systematic and complete annotation of these relations is costly and challenging, due to the quadratic number of event pairs that need to be considered. Consequently, many current event relation datasets lack systematicity and completeness. In response, we introduce *EventFull*, the first tool that supports consistent, complete and efficient annotation of temporal, causal and coreference relations via a unified and synergetic process. A pilot study demonstrates that EventFull accelerates and simplifies the annotation process while yielding high inter-annotator agreement.

1 Introduction

Identifying the semantic relations between events mentioned in a text, notably temporal, causal and coreference relations, has been a fundamental goal in NLP. Substantial efforts have been devoted to developing various datasets that capture some or all of these relations (O'Gorman et al., 2016; Hong et al., 2016; Wang et al., 2022). These datasets were then leveraged to develop and to evaluate corresponding models for detecting event-event relations. The output of such models has been utilized in a range of downstream applications, with recent examples including event forecasting (Ma et al., 2023), misinformation detection (Lei and Huang, 2023), and treatment timeline extraction (Yao et al., 2024), among others.

Models for detecting event relations are expected to produce *complete* output, that is identifying *all* event relations that can be inferred from the given text. Accordingly, such models should ideally be evaluated, and trained, on datasets in which event relation annotation is in itself complete, in the sense that each pair of targeted events has been classified for its potential relationships. Unfortunately, though, exhaustive manual annotation of all event-event relations is typically considered extremely challenging or impractical, since the number of event pairs to be considered is quadratic in the number of targeted event mentions in the text (Naik et al., 2019). As the number of event mentions grows, this task quickly becomes both too time consuming and cognitively unmanageable.

Faced with this inherent annotation complexity, many datasets adopted an annotation protocol that restricts the number of events or event pairs considered for annotation through various restrictions and heuristics. Notably, TB-Dense (Chambers et al., 2014), ECB+ (Cybulska and Vossen, 2014), MEANTIME (Minard et al., 2016), and EventStoryLine (Caselli and Vossen, 2017) restrict event pairs to a span of two consecutive sentences. This limitation inherently prevents testing and training models on longer-range relations. Other datasets, such as TimeBank (Pustejovsky et al., 2003b) and MAVEN-ERE (Wang et al., 2022), did not publish a systematic annotation execution protocol that guarantees actual complete annotation, and were subsequently criticized for being incomplete in their relation annotation (Pustejovsky and Stubbs, 2011; Rogers et al., 2024). Further, some researchers aimed to avoid the cost of manual annotation altogether and employed fully- or partly-automatic dataset creation methods (Mirza et al., 2014; Madaan and Yang, 2021; Alsayyahi and Batista-Navarro, 2023; Tan et al., 2024). These approaches inherently incorporate biases introduced by the employed automated method, making them less reliable for testing purposes while being prone to yield biased models when used for training. Finally, motivated by similar observations to ours, the recent NarrativeTime project (Rogers et al., 2024) does emphasize relation annotation completeness, supported by a corresponding annotation tool. However, their work addresses only temporal

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations), pages 494–508

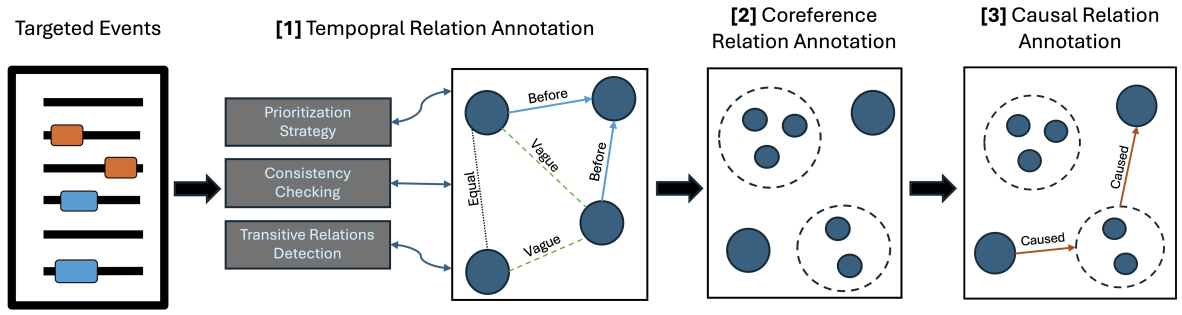


Figure 1: The EventFull annotation pipeline begins with a document containing marked targeted events and proceeds through three stages: [1] **Temporal Relations:** annotators establish temporal relations between event pairs, supported by three processes, including pair prioritization strategy, consistency checking, and transitive relation detection (§3.2); [2] **Coreference:** annotators identify coreferring event mentions; [3] **Causal Relations:** annotators determine causal relations for pairs of events.

relations, while employing a complex annotation scheme that requires expert annotators. Indeed, their actual annotation was performed by two of the authors, overall limiting the replicability of this approach.

In this paper, we aim to close major gaps in prior annotation protocols. To that end, we introduce a simple-to-use annotation tool that facilitates future creation of event relation datasets (for covering multiple relation types, new genres and languages, etc.), which *guarantees* complete relation annotation . Our web-based tool, *EventFull* (Figure 1), fulfills four critical goals, which constitute our primary contributions: (1) supporting joint synergetic annotation of the three prominent event-event relation types: temporal, causal, and coreference; (2) for any given set of targeted events within the text, the gold output relations are guaranteed to be *complete*, classifying all event pairs for the three relation types; (3) supporting efficient annotation while guaranteeing annotation consistency, via automated transitive completion and consistency checks, while leveraging constraints imposed across the three relation types (see §3.2); (4) ease of use by a non-expert annotator, lowering the bar for future dataset creation. To the best of our knowledge, EventFull is the first available tool that effectively supports and integrates all these goals.

In the remainder of the paper, we first survey relevant background and related work (§2), followed by the presentation of EventFull's input and output structure and its workflow (§3). Finally, we present a pilot study in which we demonstrate the effectiveness of EventFull (§4). The tool and code are publicly available at `https://github.com/AlonEirew/EventFull`.

2 Background and Related Work

This section first provides relevant background regarding event relations, followed by a short survey of prior annotation tools.

2.1 Event Relations

We focus on the three event-event relations which have been addressed most broadly in the literature: **temporal** (Do et al., 2012) — identifying the temporal order between events; **coreference** (Raghunathan et al., 2010) — indicating whether two event mentions in the text refer to the same real-world event; and **causal** (Mirza et al., 2014) — detecting whether one event caused another event to happen.[1]

Event relations satisfy various constraints. Temporal relations, which connect *any* two events, must maintain transitive consistency (Verhagen, 2005): if event A precedes B and B precedes C, then A necessarily precedes C (Allen, 1983). Temporal order induces further constraints on the other two relations. In a causal relation, the causing event must *precede* the other (Caselli and Vossen, 2017), while coreferring event mentions must temporally *co-occur* (Cybulska and Vossen, 2014). As described in §3.2, we leverage these constraints in our tool to increase annotation efficiency while guaranteeing its consistency.

Temporal and causal relations have been modeled at varying levels of granularity. The number of temporal relation types ranged from 13-14 in Allen's interval algebra (Allen, 1984) and TimeML (Pustejovsky et al., 2003a) to 4-6 in recent ap-

[1]Another relation type often considered is *sub-event* (Glavaš et al., 2014), which falls beyond the scope of this work.

proaches (Chambers et al., 2014; Ning et al., 2018b; Wang et al., 2022), aiming to increase annotation consistency and scalability. Fine grained causal relations distinguish three sub-types — *cause*, *precondition*, and *prevention* (O'Gorman et al., 2016; Caselli and Vossen, 2017), while other approaches model only a single *cause* relation (Do et al., 2011; Ning et al., 2018a), which is easier to annotate and model. Adopting fine-grained relations has proven challenging even for expert annotators, as evidenced by low inter-annotator agreement (Hong et al., 2016; O'Gorman et al., 2016). To support annotation by non-experts, we focus on the more coarse-grained relation types (§3.1).

Finally, we note that various methods have been proposed for determining the set of event mentions over which event relations will be annotated. These include considering all mentions (O'Gorman et al., 2016), only verbal mentions (Chambers et al., 2014), only actually-occurring events (Ning et al., 2018b; Wang et al., 2022), or salient events in the text (Madaan and Yang, 2021; Tan et al., 2024). To allow flexibility in the selection of events, we leave event mention selection to be performed independently by dataset creators as a preprocessing step for EventFull (§3.1).

2.2 Prior Annotation Tools

Event relation annotation has been carried out by two types of tools. The first involves adapting *general-purpose* annotation tools, such as BRAT (Stenetorp et al., 2012) and CAT (Bartalesi Lenzi et al., 2012). However, these general tools do not leverage specific properties of event relations, as mentioned above in §2.1, to support annotation completeness, consistency and efficiency. In contrast, targeted annotation tools have been developed to support the annotation of individual relations, including for temporal (Derczynski et al., 2016; Rogers et al., 2024) and coreference relations (Bornstein et al., 2020). Some of these tools are rather complex, suitable for expert annotators. EventFull, on the other hand, supports synergetic annotation of all the primary three event relation types, suitable for non-expert annotators, and provides targeted automated support for completeness, consistency and efficiency.

3 The EventFull Annotation Tool

As described in §1, EventFull guarantees *complete* and *consistent* annotation for temporal, causal and coreference relations between events mentioned in an input text, while minimizing the manual annotation effort. We next describe its input and output structure, followed by a detailed description of its functionality and workflow.

3.1 Input and output

Input EventFull receives as input a text, marked with targeted event mentions to be covered by the event relation annotation. As mentioned earlier (§2.1), there are many different approaches for detecting event mentions in a text and for selecting those for which relations will be annotated. Therefore, we intentionally leave the detection and selection of event mentions orthogonal to EventFull, leaving dataset creators the flexibility to choose their own methods for this preprocessing step. Thus, EventFull focuses on the challenging task of producing a complete event relation annotation across all input event mentions.

Nevertheless, as an optional auxiliary functionality, EventFull allows annotators to review the set of marked event mentions in the text,[2] enabling them to exclude certain mentions from the subsequent relation annotation process (illustrated in Appendix F, Figure 3).

Output As discussed in §2.1, in order to support simplified and consistent annotation, we focus on the most prominent relation classes for each relation type. For temporal relations, EventFull adopts the *before*, *after*, *equal*, and *uncertain* relations, following the MATRES dataset (Ning et al., 2018b). For coreference, we annotate the *coreference* relation, as defined in ECB+ (Cybulska and Vossen, 2014). For causal relations, we focus exclusively on the *cause* relation, consistent with Do et al. (2011) and Ning et al. (2018a).

Accordingly, the annotation output first specifies a set of event *coreference-clusters*, where each cluster includes a set of event mentions (possibly a singleton) that refer to the same real-world event, hence providing a representation for that event. Then, each ordered pair of coreference clusters is associated with a *temporal* relation, and, if applicable, with a *cause* relation. Importantly, the annotation process guarantees that each pair of event mentions has been classified by the coreference relation, while each pair of events (coreference clusters) has been classified for their temporal and

[2]These can be extracted using any off-the-shelf event mention detection tool.

causal relation.

3.2 Relation Annotation Workflow

EventFull employs a layered annotation approach, dividing the task into three subsequent sub-tasks (screenshots shown in Appendix F), namely temporal, coreference and causal relation annotation, in this order. Each step builds upon the previous one, while leveraging the temporal constraints imposed across the relations (discussed in §2.1). Each task is supported by guidelines and instructions provided within the tool to assist annotators in performing the tasks (see examples in Appendix G). At any stage, annotators can save their progress or revise prior annotations. Once all tasks are completed the system checks for overall annotation completeness and annotations are exported. Next, we provide a detailed description of these steps.

3.2.1 Temporal Relation Annotation

The first step aims to establish a temporal relation for *all* pairs of input event mentions (see Figure 4 in Appendix F for illustration).[3] Annotators are presented with each pair of event mentions at a time, with the pair under scrutiny highlighted in its text context. Alongside the text, a graph visualization is provided, with the scrutinized pair connected by an emphasized red edge. Annotators must choose one of four label options for the edge connecting the two events, corresponding to the set of EventFull's temporal relations (specified in §3.1): *before*, *after*, *equal*, or *uncertain* — when the temporal relation cannot be inferred from the text. Notably, the graph visualization updates with each selection, aiding in tracking progress and providing flexibility by allowing annotators to directly select and annotate (or revise) the relation for event mention pairs by clicking on the corresponding pairs of graph nodes.

To ensure complete and consistent annotation while reducing manual annotation complexity, EventFull incorporates three independent background processes to monitor and adjust annotations (Figure 1): (1) a *transitive closure algorithm* (Allen, 1984), inspired by the temporal constraints of Ning et al. (2018a) (see Appendix E, Table 4), is applied after each annotation. This algorithm identifies and *auto*-annotates yet not-annotated pairs whose links can be transitively deduced from already annotated pairs (see example in Figure 2), thereby reducing the number of pairs that require manual annotation;(see Appendix B for details.) (2) To enhance the effectiveness of the above transitive closure algorithm, we leverage the observation that the *before* relation is the most frequent in prior temporal datasets (Kishore and He, 2024). To that end, we developed a *prioritization strategy* that simulates a Depth-first style search to determine the next pair to present to the annotators, starting with the first event in the text (as illustrated in Figure 2). This process is triggered after each relation annotation, streamlining the annotation process while enhancing the utility of the auto-discovery of transitive relations (see Appendix C for details). (3) To ensure annotations are free of conflicts (consistent), a *consistency checking* algorithm is applied right after the transitive closure algorithm, detecting any conflicts that arise, for example — annotating the relation for *'accident-damage'* as anything other than *before* (Figure 2). In such cases, the tool highlights the detected conflicts and notifies the annotator, pointing at the path that led to the conflict for review (see Appendix D for details).

Once all event mention pairs are annotated without conflicts, a notification confirms that no additional pairs require annotation.

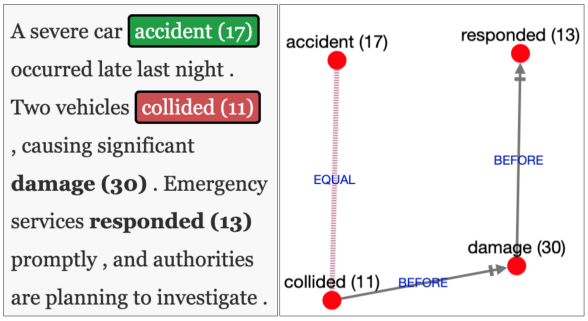


Figure 2: A simple example illustrating the prioritization strategy and automatic annotation of transitive relations. The prioritization strategy incrementally presents the pairs *'accident-collided'*, *'collided-damage'*, and *'damage-responded'* based on the relations that the annotator selected in each turn. The transitive relations *'accident-damage'*, *'accident-responded'*, and *'collided-responded'* are detected automatically.

3.2.2 Coreference Annotation

In this step annotators are asked to identify coreferring pairs of event mentions (which refer to the same real-world event; See Figure 5 in Appendix F for illustration). Inherently, this step requires considering only pairs of event mentions whose tempo-

[3]Notice that while temporal relations would eventually be induced at the level of coreference clusters, it is overall more efficient to conduct the temporal annotation in the first stage, at the event mention level (see §3.2.2 and Table 3).

ral relation is *equal* (co-occurring mentions), which immensely reduces the number of pairs that need to be considered (see Table 3), while leveraging the preceding temporal annotation step.

This step is guided by the tool, which presents to the annotator a targeted event at a time, proceeding by the text order. For each targeted event, the tool highlights all events that temporally co-occur with it, in both the text as well as in a graphical visualization of all event mentions. The annotator then selects, out of these co-occurring mentions, all those that are judged to corefer with the targeted mention. This selection defines a coreference cluster, and the process iterates for the next available mention in the text. Upon completion, event mentions not included in any cluster are marked as singleton clusters. Additionally, a discrepancy detection algorithm ensures consistency by notifying annotators of any event mention that was linked to distinct clusters, asking them to resolve the conflict.

It should be noted at this point that the consistency checks in the temporal annotation phase (Table 4) guarantee that the temporal relations for all co-occurring mentions would be identical, which in turn applies to all mentions in a corefernce cluster. This induces a consistent temporal relation annotation at the level of coreference clusters.

3.2.3 Causal Relation Annotation

Given the coreference clusters from the previous step, each representing a single real-world event, causal relation annotation now considers pairs of events rather than pairs of event mentions. For this step we assessed two prior annotation flows in the literature. The first is the RED protocol (O'Gorman et al., 2016), which requires considering independently a causal relations for each pair of events at a time. The second is the EventStoryLine methodology (Caselli and Vossen, 2017), where an event is presented alongside *all* its temporally preceding events, out of which the annotator is asked to identify all causing events. As we found in our preliminary experiments, this latter approach is faster, reduces cognitive load and supports higher-quality annotation (details provided in Appendix A), and hence we adopted it for our annotation flow.

A screenshot of causal relation annotation is illustrated in Figure 6 in Appendix F. For each targeted event at a time, annotators review a representative mention of it alongside representative mentions of all its preceding events, which are highlighted in the text and in a visualized graph representation, similar to the previous steps. The annotator is then asked to select which of the preceding events caused the current targeted one, and proceeds to the next targeted event. Following the EventStoryLine methodology, causal relations are considered as independently localized for each pair of events, without induced transitivity, hence not requiring transitive consistency checks.

4 Pilot Study

To evaluate EventFull's effectiveness, we conducted a small-scale study to assess the time, effort and quality of the resulting annotations. This section first describes the annotation procedure (§4.1) and then reviews the quality and effectiveness of the annotation process (§4.2).

4.1 Annotation Procedure

For the user study, we hired three non-expert annotators, all native English speakers and either first-degree students or graduates. They underwent three training iterations, each on a single document. Annotators were instructed to follow the methodology used in creating the MATRES dataset (Ning et al., 2018b) for selecting events and annotating temporal relations. This approach was chosen for its simplicity, making it suitable for non-expert annotators. MATRES distinguishes event mentions by their temporal characteristics — events that are actual or "anchorable in time" (e.g., they *won* the game) are included, while wishful, intentional, or conditional events (e.g., I wish they *win* the game) are excluded. Temporal relations are determined only based on the starting times of the events. For coreference and causal relations, annotators adhere to the tool's workflow (§3.2).

The annotation process was conducted on six news documents, each approximately 500 words long, where each document was annotated by all three annotators for measuring agreement. To extract the initial set of event mentions, we applied the event detection method proposed by Cattan et al. (2021), which identified an average of 60 event mentions per document. Annotators then identified "anchorable" event mentions using the tool's event selection step, averaging 35 events per document. To further refine the set and manage the study's scope, we followed prior approaches (§2.1) by instructing annotators to select the 16–18 most salient events. Following this selection, annotators proceeded to annotate the three event re-

	Temporal(κ)	Coreference(B^3)	Causal(κ)
A and B	0.72	0.98	0.83
A and C	0.75	0.93	0.77
B and C	0.68	0.95	0.74
Average (EventFull)	0.72	0.96	0.78
MAVEN-ERE	0.68	0.91	0.7

Table 1: **Agreement between Annotators:** For temporal and causal relations, the *kappa* coefficient (Fleiss and Cohen, 1973) was used, calculated using scikit-learn (Buitinck et al., 2013). For coreference, the *B-Cubed* F1 score (Bagga and Baldwin, 1998) was applied. A/B/C represent the three annotators. Additionally, for comparability, we report the agreement values from the recent MAVEN-ERE dataset (Wang et al., 2022).

	A	B	C	Average Time
Temporal	45	38	50	44.3
Coreference	8	5	10	7.7
Causal	16	15	21	17.3
Total Time (min)	69	58	81	69.3

Table 2: **Annotation Time (in minutes):** The average time taken by each annotator to complete each task for a single document. A/B/C represent the three annotators.

Relation Types	A	B	C	Average Reduction
Temporal	56.7	54.6	65.6	56%
Coreference	4.5	4.5	4.8	96%
Causal	79.2	79.2	79.2	41%

Table 3: **Annotation Steps Made:** For each document, we calculated the average number of pairs requiring classification to obtain a complete annotation for all event mention pairs (excluding symmetric ones), which averaged 136 pairs per document. We then measured the average number of pairs per document that EventFull presented for judgement to each annotator (A/B/C). The **Average Reduction** represents the percentage by which EventFull reduced the number of pairs from the total average of 136 pairs per document.

lation types. Over this process, we measure inter-annotator agreement, annotation time, and the number of annotation steps for each task.

4.2 Analysis

The final annotated set included 102 event mentions over the 6 documents, averaging 17 mentions per document, resulting in 816 distinct annotated pairs of event mentions, with an average of 136 mention pairs per document. In comparison, the prominent MATRES temporal relation dataset (Ning et al., 2018b) contains an average of 22 event mentions per document but only 50 annotated mention pairs per document.

Notably, as shown in Table 1, agreement between annotators is high across all three relations and comparable to the recent MAVEN-ERE dataset, indicating that the pilot reliably reflects a data annotation process on par with other datasets.

We also observe in Table 2 that the annotation of temporal relations roughly takes 44 minutes to complete, which is significantly more demanding than for the other two types of relations. This finding is reasonable, as temporal relation annotation requires classifying each relation into one of four classes. In contrast, coreference and causal relations involve identifying connections within a set of events relative to a focal event and its temporal context (*equal* for coreference and *after* for causal). Importantly, Table 3 indicates that EventFull assists in significantly reducing complexity for all three relation types.

5 Conclusion

In this paper we introduced EventFull, a novel tool for the end-to-end annotation of *complete* and *consistent* event relation over targeted events in a text, integrating temporal, coreference, and causal relations into a unified workflow. Starting with temporal relations, we leverage their properties to reduce the manual annotation workload and ensure consistency. Annotating temporal relations first allows the coreference step to focus only on *equal* relations and the causal step on *before* relations, thereby reducing the set of pairs requiring consideration in each step. Additionally, annotating coreference before causal relations enables all coreferring event mentions to be treated as a single event, further reducing annotation complexity. Last, we assessed the utility of EventFull through a pilot study, measuring annotation complexity, data quality via inter-annotator agreement (IAA), and annotation time efficiency. Results demonstrate that EventFull significantly reduces annotation complexity while maintaining high IAA. We hope that EventFull will facilitate the creation of new datasets that complement existing resources, focusing on constructing complete event relation annotation for targeted events in a text.

Acknowledgments

This work was supported by the Israel Science Foundation (grant no. 2827/21), and by funding from the Israeli Planning and Budgeting Committee (PBC).

References

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843.

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

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

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 79–85, Montreal, Quebec, Canada. Association for Computational Linguistics.

Valentina Bartalesi Lenzi, Giovanni Moretti, and Rachele Sprugnoli. 2012. CAT: the CELCT annotation tool. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 333–338, Istanbul, Turkey. European Language Resources Association (ELRA).

Ari Bornstein, Arie Cattan, and Ido Dagan. 2020. CoRefi: A crowd sourcing suite for coreference annotation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 205–215, Online. Association for Computational Linguistics.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake Vanderplas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. Api design for machine learning software: experiences from the scikit-learn project. *Preprint*, arXiv:1309.0238.

Tommaso Caselli and Piek Vossen. 2017. The event StoryLine corpus: A new benchmark for causal and temporal relation extraction. In *Proceedings of the Events and Stories in the News Workshop*, pages 77–86, Vancouver, Canada. Association for Computational Linguistics.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2021. Cross-document coreference resolution over predicted mentions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5100–5107, Online. Association for Computational Linguistics.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense Event Ordering with a Multi-Pass Architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4545–4552, Reykjavik, Iceland. European Language Resources Association (ELRA).

Leon Derczynski, Jannik Strötgen, Diana Maynard, Mark A. Greenwood, and Manuel Jung. 2016. GATE-time: Extraction of temporal expressions and events. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3702–3708, Portorož, Slovenia. European Language Resources Association (ELRA).

Quang Do, Yee Seng Chan, and Dan Roth. 2011. Minimally supervised event causality identification. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 294–303, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Quang Do, Wei Lu, and Dan Roth. 2012. Joint inference for event timeline construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 677–687, Jeju Island, Korea. Association for Computational Linguistics.

Joseph L. Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and Psychological Measurement*, 33(3):613–619.

Goran Glavaš, Jan Šnajder, Marie-Francine Moens, and Parisa Kordjamshidi. 2014. HiEve: A corpus for extracting event hierarchies from news stories. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3678–3683, Reykjavik, Iceland. European Language Resources Association (ELRA).

Yu Hong, Tongtao Zhang, Tim O'Gorman, Sharone Horowit-Hendler, Heng Ji, and Martha Palmer. 2016. Building a cross-document event-event relation corpus. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 1–6, Berlin, Germany. Association for Computational Linguistics.

Sindhu Kishore and Hangfeng He. 2024. Unveiling divergent inductive biases of LLMs on temporal data. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 220–228, Mexico City, Mexico. Association for Computational Linguistics.

Yuanyuan Lei and Ruihong Huang. 2023. Identifying conspiracy theories news based on event relation graph. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9811–9822, Singapore. Association for Computational Linguistics.

Yunshan Ma, Chenchen Ye, Zijian Wu, Xiang Wang, Yixin Cao, and Tat seng Chua. 2023. Context-aware event forecasting via graph disentanglement. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Aman Madaan and Yiming Yang. 2021. Neural language modeling for contextualized temporal graph generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 864–881, Online. Association for Computational Linguistics.

Anne-Lyse Minard, Manuela Speranza, Ruben Urizar, Begoña Altuna, Marieke van Erp, Anneleen Schoen, and Chantal van Son. 2016. MEANTIME, the NewsReader multilingual event and time corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4417–4422, Portorož, Slovenia. European Language Resources Association (ELRA).

Paramita Mirza, Rachele Sprugnoli, Sara Tonelli, and Manuela Speranza. 2014. Annotating causality in the TempEval-3 corpus. In *Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL)*, pages 10–19, Gothenburg, Sweden. Association for Computational Linguistics.

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

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018a. Joint reasoning for temporal and causal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2278–2288, Melbourne, Australia. Association for Computational Linguistics.

Qiang Ning, Hao Wu, and Dan Roth. 2018b. 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.

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

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 Saurí, Andrew See, Rob Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, and Marcia Lazo. 2003b. The timebank corpus. *Proceedings of Corpus Linguistics*.

James Pustejovsky and Amber Stubbs. 2011. Increasing informativeness in temporal annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 152–160, Portland, Oregon, USA. Association for Computational Linguistics.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501, Cambridge, MA. Association for Computational Linguistics.

Anna Rogers, Marzena Karpinska, Ankita Gupta, Vladislav Lialin, Gregory Smelkov, and Anna Rumshisky. 2024. Narrativetime: Dense temporal annotation on a timeline. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 12053–12073. ELRA and ICCL.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. Association for Computational Linguistics.

Xingwei Tan, Yuxiang Zhou, Gabriele Pergola, and Yulan He. 2024. Set-aligning framework for auto-regressive event temporal graph generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3872–3892, Mexico City, Mexico. Association for Computational Linguistics.

Marc Verhagen. 2005. Temporal closure in an annotation environment. *Language Resources and Evaluation*, 39(2/3):211–241.

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

Stephen Warshall. 1962. A theorem on boolean matrices. *J. ACM*, 9(1):11–12.

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.

A Causal Scheme Experiment

To address the complexity of causal annotation, we experimented with various annotation schemes, ultimately focusing on RED (O'Gorman et al., 2016) and EventStoryLine (Caselli and Vossen, 2017) methodologies. Our findings showed that annotating causality using the RED methodology required more than twice the time compared to the EventStoryLine methodology. This difference is attributed to two main factors: (1) RED requires evaluating all pairs of events, whereas EventStoryLine involves a single pass, evaluating each event with all its preceding events; (2) RED demands assessing whether the target event was "inevitable" given the source event for each pair. We observed that annotators often struggled with the concept of inevitability, even when clear causal indicators were present in the text. For instance, in the sentence, "He answered the phone *because* it rang," annotators found it challenging to evaluate causality. This led to low inter-annotator agreement, around 0.2 *kappa* (slight to fair agreement). In contrast, using the EventStoryLine methodology, annotators achieved a 0.78 *kappa*, indicating substantial agreement.

B Transitive Closure

We devised a straightforward algorithm designed to infer transitive relations. Our algorithm operates in three steps: (1) We begin with a matrix $X \in M_{n \times n}$ (n being the number of events), where each cell in the matrix represent a pair and indicates whether a direct relation has been annotated between them. (2) For every annotation made, we utilize Warsheall's algorithm (Warshall, 1962) to compute transitive closures, adapting it to accommodate the unique transitive characteristics of each relation type (as detailed in Table 4). The output of the algorithm is a matrix $\hat{X} \in M_{n \times n}$, where both direct relations and inferred transitive relations are marked. (3) The tool then examines $\hat{X}$ (excluding its diagonal) to identify the remaining pairs requiring annotation by collecting all pairs whose corresponding cells are assigned the $ANNOTATE$ relation (Table 4).

C Prioritization Strategy

Our prioritization strategy is designed to increase the likelihood of utilizing the transitive closure algorithm for automatically detecting transitive relations B. It is inspired by two key observations: (1) the *before* relation is the most frequent in prior temporal datasets (Kishore and He, 2024), and (2) story narratives naturally tend to progress chronologically as they unfold.

To that end, after applying the transitive closure algorithm (Appendix B), all pairs requiring annotation (i.e., marked with $ANNOTATE$) are arranged in the order of their appearance in the text. The first pair requiring annotation is selected, followed by examining the next pair requiring annotation. Finally, the last pair that shares the second node with this pair is selected as the next candidate pair to present to the annotator. This simple approach simulates a depth-first-like search, prioritizing yet unreached mentions. For example, in Figure 2, once the pair *'accident-collided'* is annotated as *EQUAL*, the next unhandled pair is *'accident-damage'*, and the last pair sharing the second node 'damage' is *'collided-damage'*.

D Consistency Checking Algorithm

To manage potential discrepancies, we incorporated rule-based checks into the transitive closure algorithm (Appendix B), similar to the transitive constraint logic proposed by (Ning et al., 2018a). These checks ensure that no direct annotation contradicts a newly identified transitive relation (or vice versa). At the base of our discrepancy algorithm, three nodes are considered: i, j, and k, where i and j represent our target nodes, and k represents any node through which there is a path

	$\{i,k\}$	$\{k,j\}$	$\{i,j\}$
1	BEFORE	BEFORE	BEFORE
2	BEFORE	EQUAL	BEFORE
3	EQUAL	BEFORE	BEFORE
4	AFTER	AFTER	AFTER
5	AFTER	EQUAL	AFTER
6	EQUAL	AFTER	AFTER
7	EQUAL	EQUAL	EQUAL
8	BEFORE	AFTER	ANNOTATE
9	AFTER	BEFORE	ANNOTATE
10	VAGUE	VAGUE	ANNOTATE
11	EQUAL	VAGUE	ANNOTATE
12	VAGUE	EQUAL	ANNOTATE
13	BEFORE	VAGUE	ANNOTATE
14	VAGUE	BEFORE	ANNOTATE
15	AFTER	VAGUE	ANNOTATE
16	VAGUE	AFTER	ANNOTATE

Table 4: $\{i,k\}$ and $\{k,j\}$ represent annotated paths between events i to k, and k to j. $\{i,j\}$ represents the inferred transitive relation between events i and j via k. The ANNOTATE relation indicates that this relation should be presented to the annotator for verification.

from i to j. This logic then examine whether the direct relation $\{i,j\}$ contradicts the temporal relation inferred from combining the two edges that form the path (the rules are shown in Table 4). If the direct relation $\{i,j\}$ contradicts the inferred relation $\{i,k,j\}$, EventFull will alert the annotator to resolve the contradiction. For the coreference relation, we handle cases where a user assigns a mention to one event cluster and later assigns the same mention to a different cluster. In such instances, the tool notifies the annotator about the misalignment and prompts them to verify their selection. If the annotator confirms the change, the event mention is removed from its current cluster and added to the new one.

E Temporal Constraints

Table 4 presents the temporal constraints employed in our transitivity and discrepancy detection algorithms.

F Printscreens of EventFull

We present examples of EventFull (Figures 3, 4, 5, and 6), showcasing the four annotation steps.

G EventFull Annotation Guidelines

The guidelines in EventFull are fully customizable, allowing anyone managing an annotation task to edit them to align with any annotation scheme. We present an example of the guidelines we used in our pilot study, accessible through the tool's UI (Figures 7).

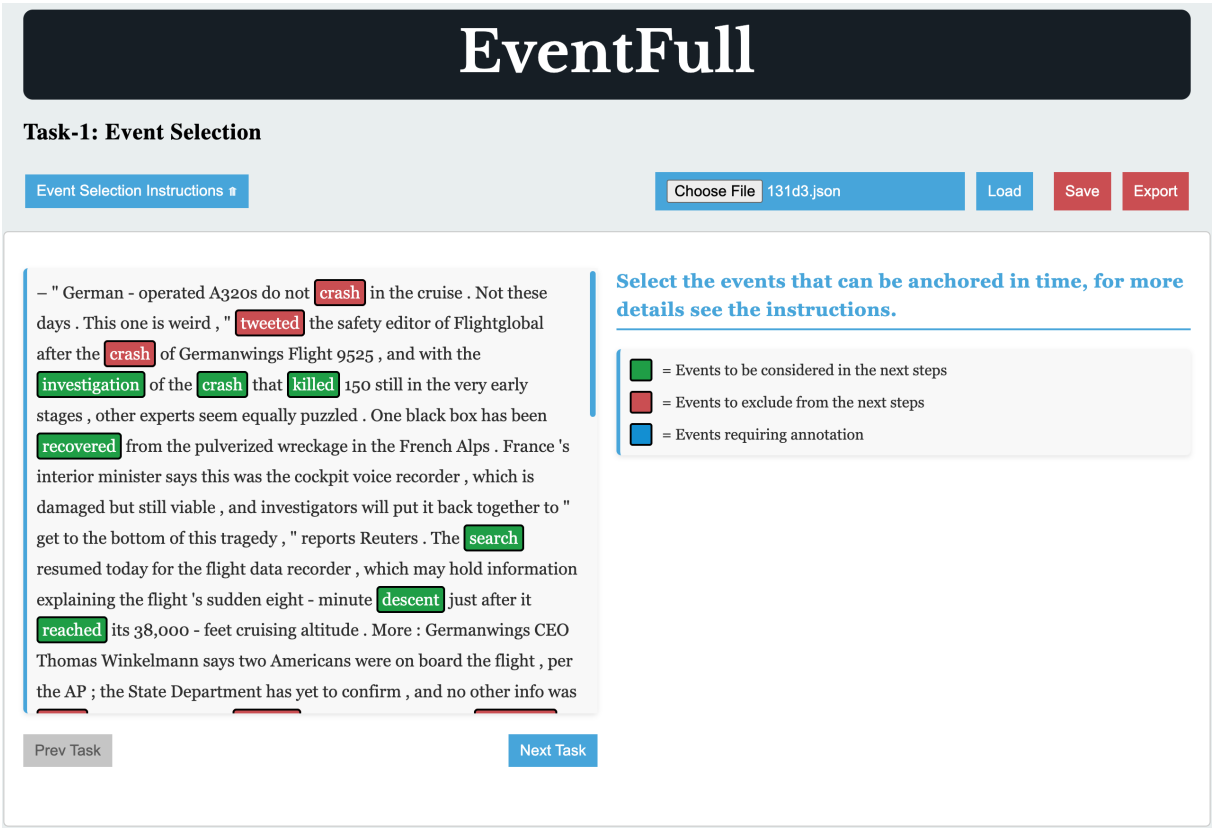


Figure 3: **Event Selection Annotation Step:** This optional step aims to refine the set of events (detailed in §3.2) by selecting the events to be considered in subsequent steps. Annotators can access guidelines by clicking the "Event Selection Instruction" button. After categorizing all events as either event or no-event, they can proceed to the next annotation task by clicking the "Next Task" button.

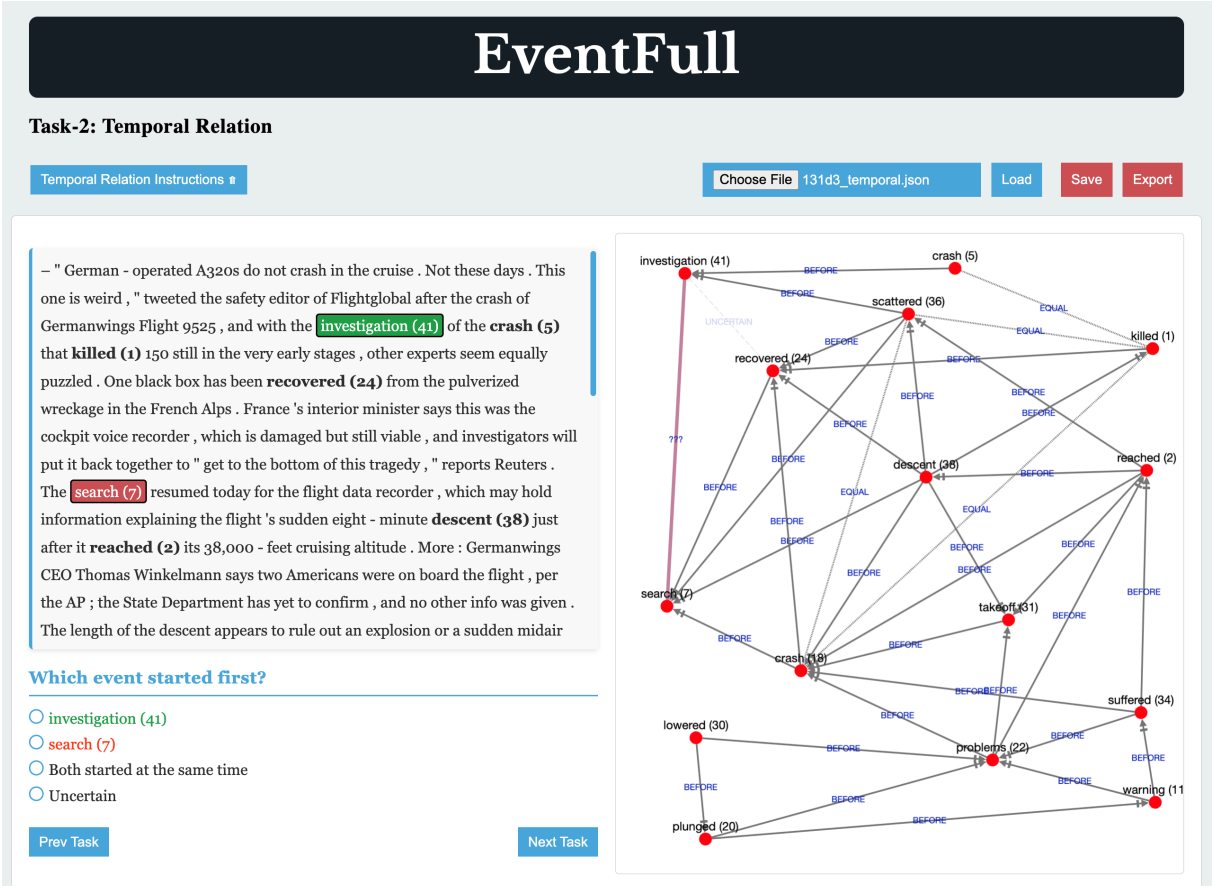


Figure 4: **Temporal Relation Annotation Step:** Annotators determine the temporal relations for each candidate pair based on the starting point of the events by selecting the appropriate radio button option (detailed in §3.2). Events requiring annotation are highlighted in green and red within the context, with the selected relation displayed in red in the visualization graph. All other event mentions are marked in bold within the context. The graph visualization tracks annotators' progress and allows manual selection by clicking on two nodes representing events in the graph. Annotated relations are displayed in the graph with their corresponding temporal label. Each event mention is assigned an identifier in parentheses, visible in both the context and graph, to facilitate locating events.

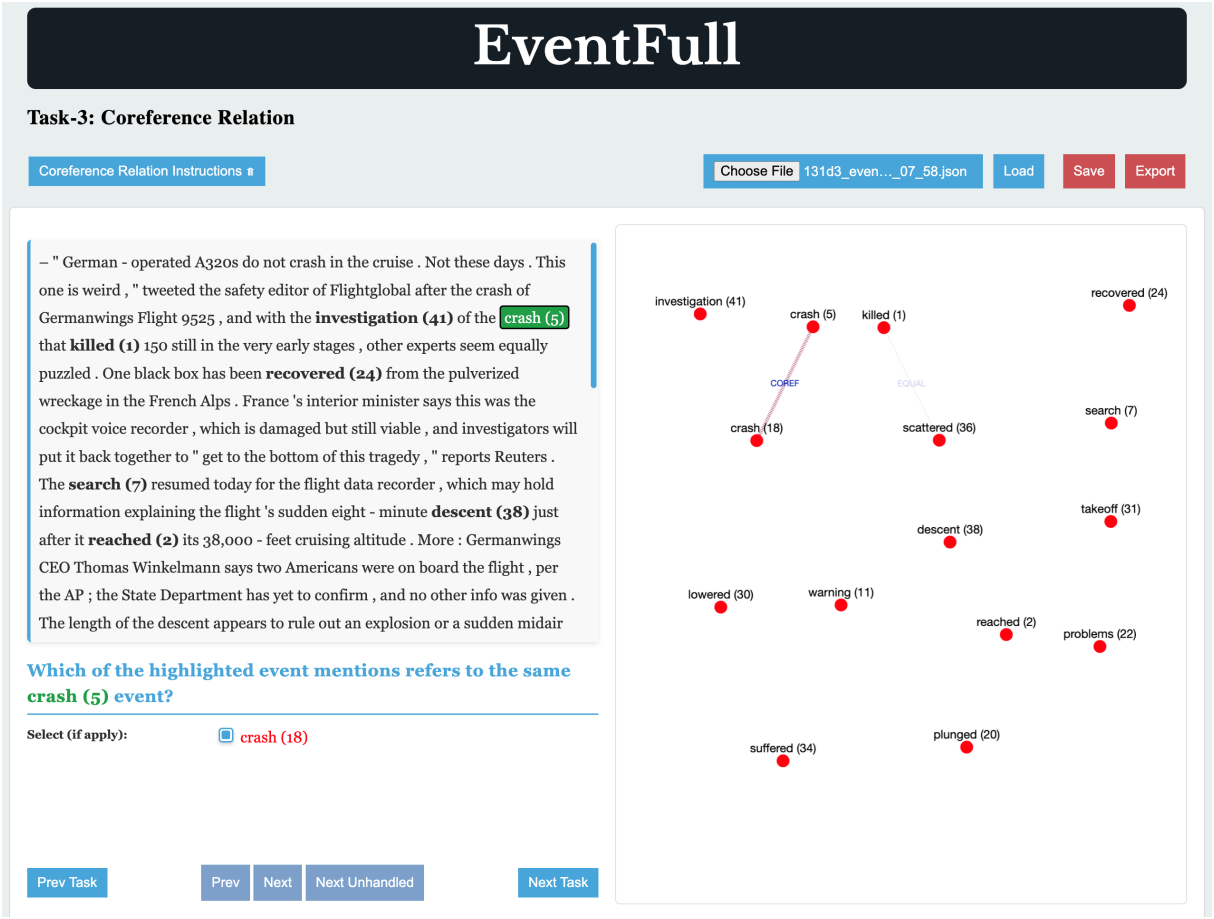


Figure 5: **Coreference Relation Annotation Step:** Annotators determine coreference relations among all candidates annotated in the *temporal* step as having an *equal* time relation. The event mention representing the event cluster is highlighted in green, while all candidate event mentions (sharing an equal time with it) are highlighted in red. Annotators are required to check the checkbox of the red event mentions that corefer with the green one and proceed to the next group using the "Next Unhandled" button. A graph visualization tracks annotators' progress, displaying only relevant relations for this step (i.e., equal and coreference). The relations under scrutiny are highlighted in red, while not-corefer relations appear faded.

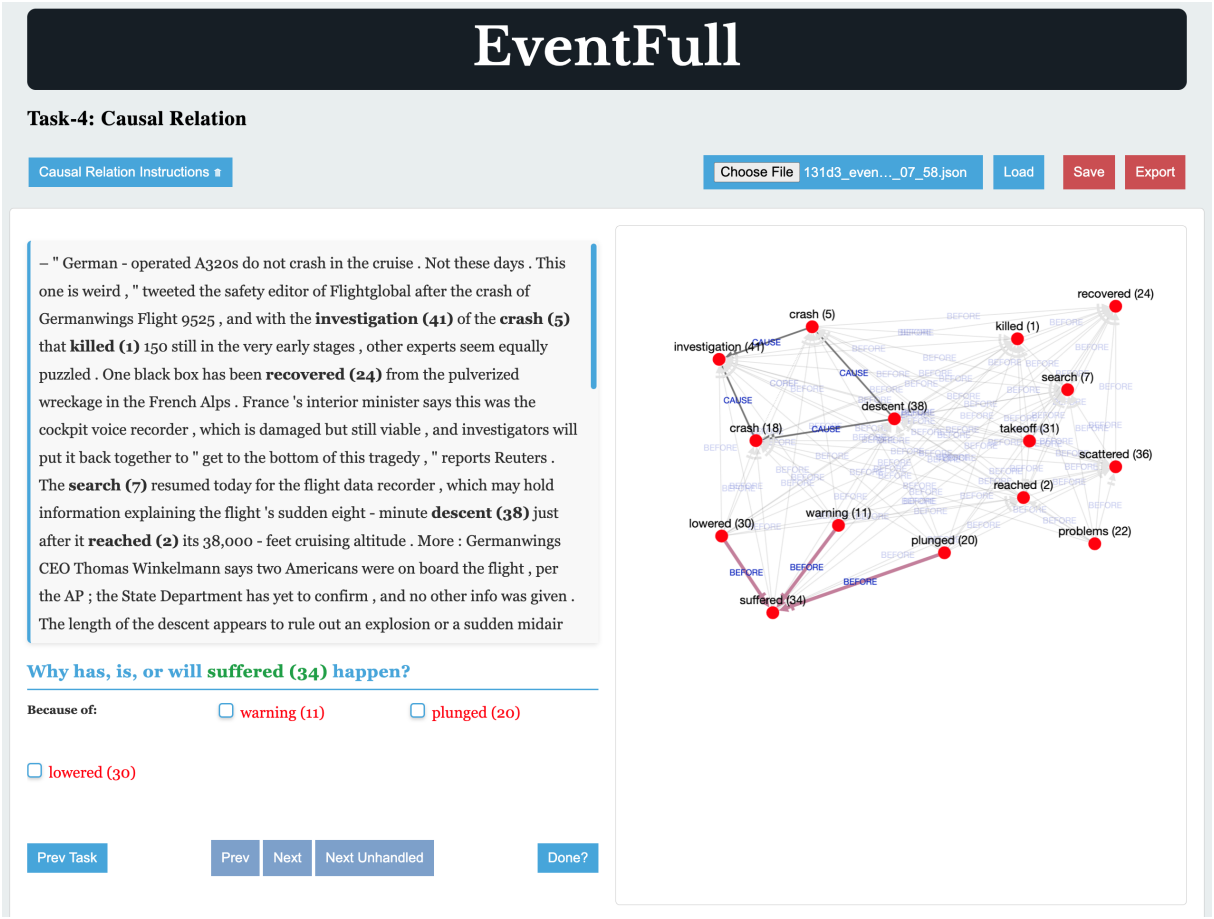
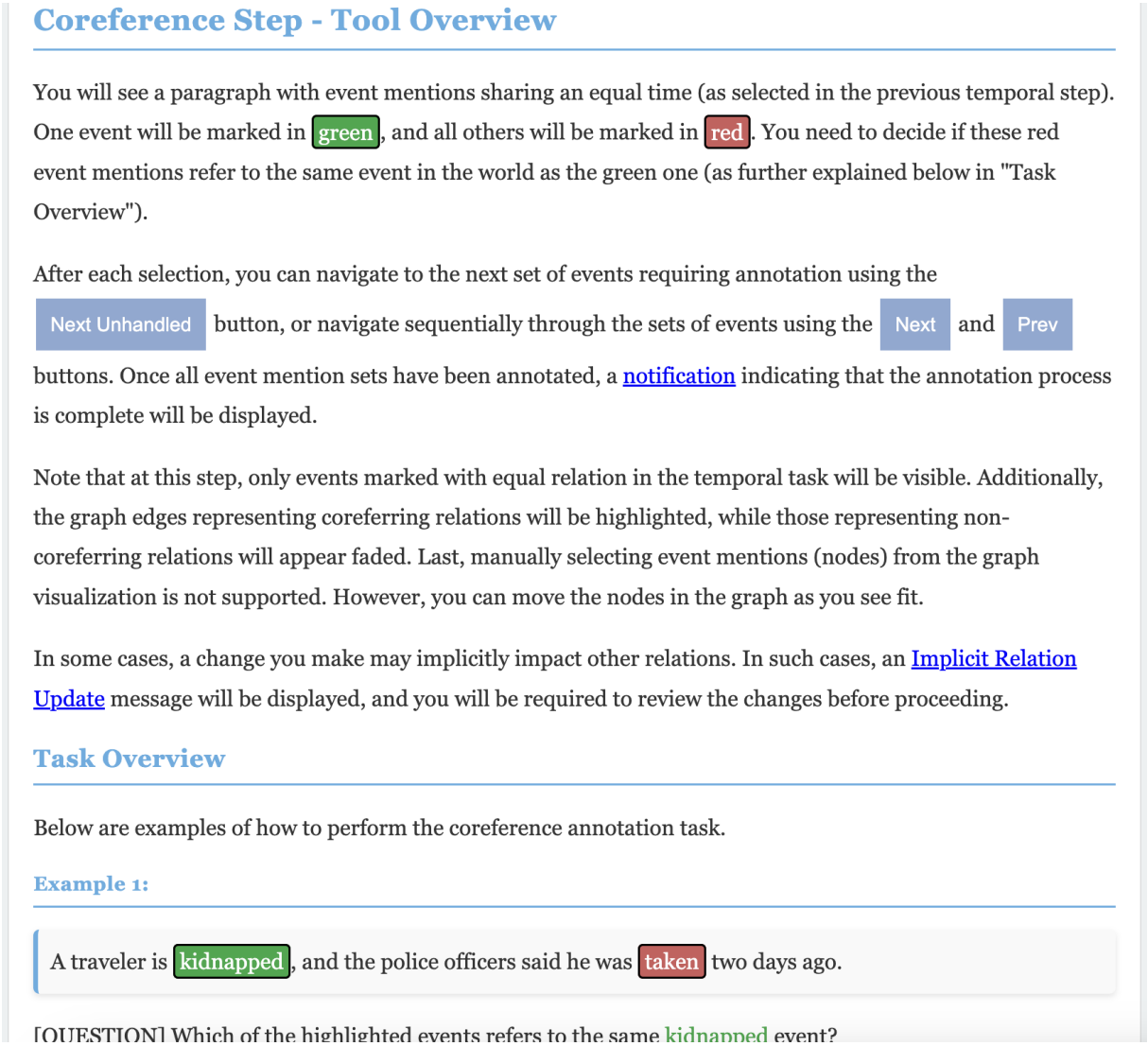


Figure 6: **Causal Relation Annotation Step:** Annotators determine causal relations among all candidates annotated in the *temporal* step as having a *before* time relation. The event mention in focus is highlighted in green, while all preceding event mentions are highlighted in red. Annotators are required to check the checkbox of the red event mention(s) that caused the green one to occur and proceed to the next group using the "Next Unhandled" button. A graph visualization tracks annotators' progress, displaying only relevant relations for this step (i.e., BEFORE and CAUSE). The relations under scrutiny are highlighted in red, with all relations already annotated as CAUSE clearly visible, while non-causal relations appear faded.



Coreference Step - Tool Overview

You will see a paragraph with event mentions sharing an equal time (as selected in the previous temporal step). One event will be marked in green, and all others will be marked in red. You need to decide if these red event mentions refer to the same event in the world as the green one (as further explained below in "Task Overview").

After each selection, you can navigate to the next set of events requiring annotation using the Next Unhandled button, or navigate sequentially through the sets of events using the Next and Prev buttons. Once all event mention sets have been annotated, a notification indicating that the annotation process is complete will be displayed.

Note that at this step, only events marked with equal relation in the temporal task will be visible. Additionally, the graph edges representing coreferring relations will be highlighted, while those representing non-coreferring relations will appear faded. Last, manually selecting event mentions (nodes) from the graph visualization is not supported. However, you can move the nodes in the graph as you see fit.

In some cases, a change you make may implicitly impact other relations. In such cases, an Implicit Relation Update message will be displayed, and you will be required to review the changes before proceeding.

Task Overview

Below are examples of how to perform the coreference annotation task.

Example 1:

> A traveler is kidnapped, and the police officers said he was taken two days ago.

[QUESTION] Which of the highlighted events refers to the same kidnapped event?

Figure 7: An example of the guidelines used in the pilot study, accessible by clicking the “Coreference Relation Instructions” button in the Coreference annotation step. Similarly, guidelines are available for the other steps within the tool, allowing annotators to revisit them as needed while performing the annotation task.