Set-Aligning Fine-tuning Framework for Auto-Regressive Event Temporal Graph Generation

Anonymous ACL submission

Abstract

Event temporal graphs have been shown as convenient and effective representations of complex temporal relations between events in text. While traditional methods are based on a pipeline approach, i.e., event extraction and relation classification, the recently proposed contextualised graph generation methods have shown promising results by employing pre-trained language models to generate linearised graphs as autoregressively. However, this inevitably led to sub-optimal graph generation as the linearised graphs exhibit set characteristics which are instead treated sequentially by language models. This is due to their conventional text generation objectives which end up mistakenly penalizing correct predictions only because of the misalignment between elements in text. In this work, we reformulate the task as a conditional set generation problem, proposing a Set-aligning Fine-tuning Framework allowing smooth employment of large language models. A comprehensive experimental assessment has shown that our proposed framework significantly benefits the event temporal graph generation, and outperforms existing baselines. We further demonstrate that under the zero-shot settings, the structural knowledge introduced through the proposed framework has a significant beneficial impact on model generalisation when the examples available are limited[1].

1 Introduction

Understanding the temporal relation between events mentioned in long documents is crucial to modelling complex text with articulated narratives. One of the widely adopted benchmarks for event temporal relation understanding is still the SemEval 2013 TempEval-3 (UzZaman et al., 2013), requiring end-to-end generation of event temporal graphs directly from raw text. An event temporal graph is a natural representation of temporal information, with the nodes representing events and the edges the temporal relationships between them, such as "*before*", "*after*", or "*simultaneous*".

Most existing approaches typically address the problem of extracting event temporal graphs through a two-step pipeline, with the first step focusing on detecting events in text, and the second step on classifying the temporal relations between them (McDowell et al., 2017; Ning et al., 2018b). However, such pipeline-based approaches suffer from well-known limitations, including (i) the need for fine-grained annotations at each step; and (ii) the potential for error propagation throughout the pipeline. In particular, in the first step, the event extractor's objective is to locate as many event triggers as possible in the given documents, leading to the inclusion of numerous trivial events that often lack relevance to the narrative and have no relation to other events. As a result, the next step for temporal relational extraction becomes burdened with many noisy events, significantly impacting the overall accuracy and efficiency of the models.

To address these limitations, Madaan and Yang (2021) introduced a reformulation of the task by generating event temporal graphs directly through conditional text generation. This approach allows for the use of large pre-trained language models and, more importantly, overcomes the typical limitations associated with the pipeline architecture. While this method involved fine-tuning a text generation model, such as GPT-2, for the generation of linearised event temporal graphs as sequences, it fails to consider an important aspect. Specifically, it does not account for the fact that the target sequence (i.e. the list of event temporal relations) is order-invariant, and should therefore be treated as a *set* rather than as an ordered sequence. For example, the following two sequences represent the same temporal graph:

[1] The experiment code will be released upon the paper's acceptance.. The human annotations will be released as span indices in agreement with the NYT corpus's license.

```
S1: [(Cuomo leaving his office, before, speak to reporters),
    ···(Cuomo leaving, before, met with representatives)]
S2: [(Cuomo leaving, before, met with representatives),
    ···(Cuomo leaving his office, before, speak to reporters)]
```

In this scenario, the conventional loss function will (mistakenly) yield a high value because most of the tokens in the corresponding positions do not match, even though the event relations are the same. This issue has a detrimental effect on the model performance for several reasons. First, it discourages the language model from generating additional edges. Generating more edges implies a greater number of potential permutations in the edge sets, making it less likely to match the target. Secondly, if the initially generated edge in the sequence differs in token count from the one in the target, it causes all subsequent edges to misalign with the target, even if they are identical, leading to a high loss value.

In this work, we propose a Set-Aligning Fine-tuning Framework (SAFF) that enables efficient employment of LLMs for auto-regressive event temporal graph generation. SAFF incorporates a novel set of losses, named Set Property Losses, along with augmented data, aimed at mitigating the challenges associated with conventional text generation loss. Using the proposed SAFF, we fine-tune language models from the T5 (Raffel et al., 2020) family with weak supervision. Additionally, we introduce the first human-annotated dataset built on the New York Times for contextualised event temporal graph generation, which we combine with existing event relation extraction datasets to evaluate the effectiveness of the SAFF framework. Experiments on the newly annotated New York Times corpus [2] show that SAFF significantly increases the number of generated edges, resulting in improved recall. Furthermore, we assess the performance of our approach on existing sentence-level event temporal relation extraction datasets, namely MATRES (Ning et al., 2018a) and TB-Dense (Cassidy et al., 2014), under zero-shot settings, and we find that the structural knowledge introduced through the proposed SAFF has an even greater impact on model generalisation when the examples available are limited.

Our contributions are three-folded:

- We introduce a model-agnostic framework, called SAFF, for event temporal graph generation. SAFF incorporates novel Set-Aligning loss functions, data augmentation, and weak supervision techniques.
- We offer a human-annotated test set and a weakly-supervised dataset specifically designed for document-level event temporal generation.
- We conduct an extensive evaluation of the fine-tuned models under various settings, demonstrating the effectiveness of the proposed framework and the potential of contextualised event graph generation.

[2] https://doi.org/10.35111/77ba-9x74

2 Related Work

2.1 Event Temporal Graph

The task of event temporal graph extraction serves as an important task for evaluating an end-to-end system which takes raw text as input and output TimeML annotations (i.e., temporal relations) (UzZaman et al., 2013). Early attempts on the task include CAEVO (McDowell et al., 2017) and Cogcomptime (Ning et al., 2018b), which relied on a combination of statistical and rule-based methods. In recent years, more efforts have been put into developing specialised sub-systems with neural network-based approaches (Ning et al., 2019; Han et al., 2019a; Tan et al., 2021a). The emergence of large language models has paved the way for end-to-end learning, treating temporal graph generation as conditional text generation (Madaan and Yang, 2021). To tackle the set misalignment issue which remained unexplored in Madaan and Yang (2021), we propose a framework based on a novel set of Set-Aligning losses, aiming at enhancing autoregressive event temporal graph generation.

It is worth noting that there is another related and more widely-recognised task called *temporal relation extraction*, which aims at classifying the type of temporal links between pre-extracted events (Wang et al., 2020; Wen and Ji, 2021; Tan et al., 2023). While Han et al. (2019b) proposed a joint extraction model for events and event temporal relations, they rely on event extraction supervision signals, which our work does not need.

2.2 Graph Generation with Language Models

Generating graphs with language models has been explored in many areas. For example, Bosselut et al. (2019) fine-tunes GPT on the ATOMIC commonsense knowledge graph (Sap et al., 2019). Melnyk et al. (2022) proposed a multi-stage system for knowledge generation based on T5. However, these studies do not generate an entire graph in one generation. In contrast, Madaan et al. (2021) generated inference graphs using a combination of a graph generator and a graph corrector for queries in defeasible reasoning. Zaratiana et al. (2023) generate entities and entity relations with an autoregressive LM, but they did not consider the set property of the target. Different from them, we focus on the set property of the generation sequence, which is particularly important in the setting where both the input document and output sequence are considerably longer.

2.3 Conditional Set Generation

Text generation models are primarily designed for generating text with strict linear orders, making them suboptimal for generating sets. This limitation has been acknowledged in recent NLP research, where efforts have been made to adapt seq2seq frameworks for tasks like multi-label classification and keyword generation (Qin et al., 2019; Ye et al., 2021). Vinyals et al. (2016) studied the general challenge of using sets as either input or target output for text generation models. They found in both cases, the order of elements in the set has a significant impact on convergence and final perplexity. This implies that there may exist an optimal order for the input or output set sequence, and they proposed allowing the model to search for this order during training. Instead of resorting to exhaustive search, Madaan et al. (2022) proposed to use data augmentation to enforce order-invariance and prepend the set's cardinality to the target sequence to ensure the correct cardinality. While previous research has tackled multi-label prediction and keyphrase generation, our work delves into the unique challenges presented by event temporal graph generation, which involves long sequences and partially ordered properties.

In a more general sense, the object detection task from computer vision also involves set prediction. They use parallel decoding to generate the elements in a set based on object queries (Carion et al., 2020; Chen et al., 2022). Tan et al. (2021b) adopted a similar approach in name entity recognition with a non-autoregressive decoder. Different from them, the set elements (event relations) in our task are not concrete spacial objects or text spans but instead are varied in length and scattered across each document. This makes object queries and non-autoregressive decoding inapplicable in our settings.

3 Set-Aligning Fine-tuning Framework

Madaan and Yang (2021) first explored the possibility of end-to-end event temporal graph generation using neural language modelling. Since then, however, this task has remained under-explored, with numerous unresolved issues. To elaborate, the first concern is that Madaan and Yang (2021) framed graph generation as a conventional sequence generation problem, whereas it is fundamentally a set generation problem. Secondly, the dataset they built primarily consists of small-sized graphs, failing to challenge the model in terms of document-level understanding. Lastly, their investigation mainly centred on GPT-2, while the landscape of LLMs has evolved with the emergence of models featuring distinct structures (e.g., encoder-decoder) and new paradigms (e.g., in-context learning) in recent years. In this study, we address these three aspects to enhance the understanding of sequence-to-sequence temporal graph generation.

Although our proposed framework is designed to be model-independent, several factors have led us to choose Flan-T5 as the base model for our experiments: (i) Based on our preliminary experiments, Flan-T5-base hits the sweet spot in terms of performance vs. resource consumption, allowing us to test more variants; (ii) its encoder-decoder structure is well-suited to document-level graph generation, due to its efficiency in processing comprehensive information in lengthy documents.

3.1 Event Temporal Graph Modelling as Edge Set Generation

An event temporal graph is a directed graph with no isolated vertex. Each edge in the graph describes a temporal relation between two events, and self-loops are not permitted. Following Madaan and Yang (2021), we represent these graphs by linearizing them into strings using the DOT graph description language (Gansner, 2006) (example shown in Figure 1). Given that event temporal graphs do not have isolated vertices, the sequence essentially represents the edge set of the graph.

We model the probability of generating a string y, which is a linearised representation of the event temporal graph G, conditioned on a document $X = (x_1, x_2, ..., x_n)$ using a language model:

$$p_{\text{LM}}(y|X) = \prod_{t=1}^{T} p(y_t|X, y_{<t}) \qquad (1)$$

where y is a string formatted in DOT notation.

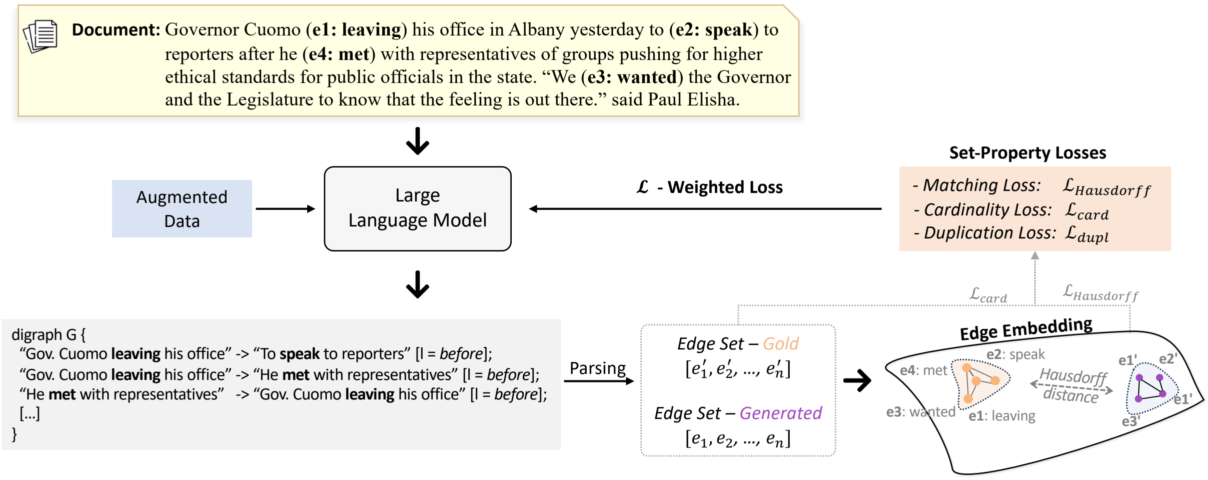

Figure 1: Set-Aligning fine-tuning framework (SAFF).

3.2 Data Augmentation

The target sequences of event temporal graph generation are essentially sets rather than strictly ordered text sequences. Therefore, conventional text generation loss can inadvertently penalize the token order and force the arrangement of elements to match the order in the target sequence, which is not necessarily the optimal order. This enforced order may lead to sub-optimal performance (Vinyals et al., 2016). A potential solution is to introduce random permutations of set elements as augmented training examples, which has already been shown effective in tasks like multi-label classification and keyphrase generation (Madaan et al., 2022). Specifically, in the context of event temporal graph generation, the elements correspond to the edges in the target string. The substrings representing the edges are randomly shuffled, while the rest of the string remains unchanged.

Prepending the set cardinality of the ground-truth edge set to the generation target may also help constrain the generation model to avoid over-generation (Madaan et al., 2022). However, such attempts in our preliminary experiment led to an approximate 4% drop in edge F_1 score, despite a significant reduction in the number of generated edges. Thus, we decided not to incorporate the cardinality into the final framework.

3.3 Set Property Losses (SPL)

Simply adding augmented data to train models does not address the fundamental issue of set alignment. Several challenges arise in this approach. First of all, it is unrealistic to add all permutations, especially when dealing with long documents containing numerous event relations, as the training data will grow at a rate proportional to the factorial of the cardinality of the target set. More importantly, with each augmented example, the loss function would still penalize the unobserved permutations of the set. This would make the training unstable.

The core challenge lies in finding an effective way to compare the linearized target graph with the linearized generated graph, without relying on a strict token-by-token comparison as in conventional text generation. To tackle this issue, we propose introducing modifications to the generation objective. As the linearized graph essentially represents the edge set of the graph, we can simplify the graph comparison problem into a set comparison problem. Our approach involves several components. Firstly, we add a set cardinality loss to encourage the model to generate an adequate number of temporal relation edges. Then, we introduce a duplication loss to penalize any repetition of elements in the edge set. Lastly, we design a set matching loss that assesses the semantic similarity between elements in the target edge set and those in the generated edge set. Collectively, the above loss functions are referred to as Set Property Losses (SPL). The SPL is integrated with the conventional token-level cross-entropy loss through a weighted average.

To compute the set property losses, a graph string needs to be first sampled from a language model given a training input. Then, this sequence is parsed into a list of edges E, where each edge e is a triplet consisting of a head event, a relation type, and a tail event (h, r, t). Now, the number of edges and duplicated edges can be counted. Let $\mathcal{E}$ denote the set of all the unique edges in E. The values for the set cardinality loss and the duplication loss can be computed as follows:

$$\mathcal{E} = \{e | e \in E\} \tag{2}$$

$$L_{\text{dupl}} = \frac{|E| - |\mathcal{E}|}{|\mathcal{E}|} \tag{3}$$

$$L_{\text{card}} = \frac{\text{abs}(|\mathcal{E}'| - |\mathcal{E}|)}{|\mathcal{E}|} \tag{4}$$

where function abs$(\cdot)$ denotes taking the absolute value, $\mathcal{E}'$ denotes the ground-truth edge set.

To compute the set matching loss, we assess the similarity between the generated set and the target set by comparing the semantic similarity of the edges across the two sets. We take the last layer of the decoder's representations of the respective tokens as the semantic representations of the events and the relation type. Then, we concatenate these representations as the semantic representation of each edge:

$$z_h = H_{[h_1, h_2, ..., h_m]} \tag{5}$$

$$z_r = H_{[r_1, r_2, ..., r_s]} \tag{6}$$

$$z_t = H_{[t_1, t_2, ..., t_n]} \tag{7}$$

$$\bar{e} = \left[\text{pool}(z_h); \text{pool}(z_r); \text{pool}(z_t))\right] \tag{8}$$

where H is the last-layer hidden states of the decoder. $[h_1, ..., h_m]$, $[r_1, ..., r_s]$, and $[t_1, ..., t_n]$ are the indices of the head event, relation type, and tail event, respectively. z_h, z_r, z_t denote the semantic representations of the head event, relation type, and tail event, respectively. pool$(\cdot)$ represents the average pooling function. $\bar{e}$ denotes the semantic representation of the edge.

We now possess two sets of embeddings: one compassing the edge embeddings extracted from the target graph, and the other containing the edge embeddings derived from the generated graph. Essentially, they can be considered as two sets of points in the representation space. Thus, we can measure the similarity of the two graphs by measuring the distance between the two point sets (manifolds) in the representation space. The Hausdorff distance, originally defined to measure the separation between two subsets within a metric space, has recently found applications in machine learning for measuring the distance between two sets of embeddings (Schutze et al., 2012; Wang et al., 2023). We compute the average Hausdorff distance as the measure:

$$d_H(\mathcal{E}', \mathcal{E}) = \frac{1}{|\mathcal{E}'|} \sum_{\bar{e'} \in \mathcal{E}'} \min_{\bar{e} \in \mathcal{E}} d_{cos}(\bar{e'}, \bar{e}) + \frac{1}{|\mathcal{E}|} \sum_{\bar{e} \in \mathcal{E}} \min_{\bar{e'} \in \mathcal{E}'} d_{cos}(\bar{e'}, \bar{e}) \tag{9}$$

where the distance of an edge pair is computed by the cosine distance $d_{cos}(\cdot)$.

3.4 Fine-tuning with Set Property Losses

Unlike the set prediction methods based on parallel decoding (Carion et al., 2020; Tan et al., 2021b), set-based objectives, such as SPL, cannot be directly used as the main objective in autoregressive generation. There are two primary reasons for this. The first reason is that obtaining the SPL requires sampling from the decoder, which would reduce the training speed significantly. Moreover, the second reason is that the language model will struggle to generate sequences in DOT format accurately because learning the token dependency for such format requires the language modelling objective. Consequently, the sequence parser will fail to recognize any valid edges within the sequence, resulting in high SPL values and hindering the training.

To avoid the problems mentioned above, we introduce the SPL after a certain number of fine-tuning iterations. Once the model has acquired a basic proficiency in generating correct DOT sequences, the SPL can function as intended. SPL can also be viewed as a regularization to prevent LLMs from overfitting to the order of the target set shown in the training samples.

We explored alternative approaches to incorporate SPL, but they reported inferior performance compared to the method eventually included in our framework. We discuss those alternative methods in the Appendix A.

4 Experiment

4.1 NYT Temporal Event Graph Dataset

There are several event temporal relation extraction datasets with pairwise event relation annotations, such as MATRES and TBD. It is theoretically possible to convert these annotations into document-level event temporal graphs. However, our preliminary experiments have shown that even when merging all of these datasets (resulting in 4,684 training documents), it is not sufficient to fine-tune a large language model to achieve acceptable performance. To address this limitation, we opted to

	NYT-train	NYT-test	NYT-human
Total documents	18,263	1,000	22
Total events	846,022	47,251	661
Node degree	2.52	2.54	2.34
Total relations	1,066,264	60,056	528
before	578,216	32,729	465
after	412,704	23,200	0
includes	7,922	450	12
is_included	41,964	2,332	0
simultaneous	25,458	1,345	51

Table 1: The statistics of the NYT temporal event graph dataset. Node degree represent the average number of relations each event has.

build a significantly larger dataset on a selection of data from the New York Times (NYT) corpus using a weak supervision approach, drawing inspiration from the work of Madaan and Yang (2021). Nevertheless, we introduced additional steps in the data selection process to ensure that the selected documents contain high-quality event temporal graphs, which were not taken in Madaan and Yang (2021).

Firstly, we performed topic modelling using Latent Dirichlet Allocation (LDA) on the MATRES and TBD datasets to extract a set of topics. Then, we identified general descriptors that are semantically similar to these topics (e.g., politics, diploma, sports, etc.). This selection process was crucial because, following training with noisy labels, our intention was to evaluate the model's performance on these datasets under zero-shot settings. We further analysed the most noteworthy events in these descriptors to ensure they were narrative-oriented, because articles that weave stories tend to contain a wealth of event temporal relations. To identify the most significant events, we employed a metric similar to TF·IDF which we could describe as "event frequency $\times$ inverse-descriptor frequency".

$$\text{ef}\cdot\text{idf} = \frac{f_{\mathfrak{e},d}}{\sum_{\mathfrak{e}'\in d} f_{\mathfrak{e}',d}} \cdot \log \frac{|D|}{|\{d \in D : \mathfrak{e} \in d\}|} \quad (10)$$

where $\mathfrak{e}$ is an event and d is a descriptor. $f_{\mathfrak{e},d}$ is the number of times that event $\mathfrak{e}$ occurs in the documents with the descriptor d. $\sum_{\mathfrak{e}'\in d} f_{\mathfrak{e}',d}$ is the total number of event occurrence in the descriptor d. $|D|$ is the total number of descriptors in the corpus. $|\{d \in D : \mathfrak{e} \in d\}|$ is the number of descriptors where the event $\mathfrak{e}$ appears.

The descriptors that are selected and the number of documents in them are listed in the Appendix D.1. After choosing the documents, we acquire the event temporal graph by running an off-the-shelf event and temporal relation extraction tool called CAEVO (McDowell et al., 2017). CAEVO is more scalable than Cogcomptime (Ning et al., 2018b), making it suitable for building a large-scale dataset.

Then, each temporal graph is represented in DOT format, and every event verb is prefixed and suffixed with its noun phrase and object, respectively. Note that we did not break the documents into short segments as Madaan and Yang (2021) did. Instead, we keep the data strictly at the document level which is a more challenging setting because the model needs to analyse the entire document and generate a much larger graph. In the dataset we built, a target graph has about 46 nodes and 58 edges on average. While in Madaan and Yang (2021), the average number of nodes is 4 and the average number of edges is 5 in a document-level event temporal graph. Moreover, their events have 1.54 relations on average, while events in our data have 2.52 relations on average, showing that the graphs in our dataset are much more complex. In practice, these complex documents are usually the ones that require analysis, and a model developed based on simpler inputs cannot handle them directly.

4.2 Human-annotated Test Data

Aside from testing with the CAEVO-created data, we recruited human annotators to annotate a test split of the NYT data. We performed a preprocessing step regarding the relation types by merging the reciprocal relations, such as transforming *after* into *before*, *is_included* into *includes* by swapping the head and tail events. For example, "I had dinner after I had lunch" is equivalent to "I had lunch before I had dinner". This processing not only streamlined the annotation process but also enhanced the model performance (refer to experimental results in Appendix C). We recruited crowd workers from Prolific[3] platform, which is a research-focused platform providing verified human workers. We recruited 24 participants in total (including pilot testing runs). To make sure the participants can understand and annotate the article efficiently, we only recruited native English speakers who have an education level higher than High school diploma/A-levels. We put 4 documents, which are randomly sampled from the same descriptor set as the training and testing of the selected NYT corpus, into each unit task. There is a shared document across all the tasks to compute the inter-annotator agree-

[3]prolific.com

ment (IAA). To minimize discrepancy, we asked 2 participants to first identify the event triggers in each unit task. We then merged the event annotations from the participants by taking the union of the spans (if there are overlapped spans, we take the longer span). Then, we asked another participant to annotate the event temporal relation based on the identified events. We also included the outputs from the CAEVO model to serve as examples, but we explicitly asked the participants to correct the annotations by adding, removing, or changing the CAEVO's annotations. In the end, we collected 22 documents as the human-annotated test set. On the event identification, we compute IOU (Intersection over Union) as a measure of agreement between the annotators. Average across 7 tasks, the IOU between the event spans is 0.8986. For the relation annotations, we compute the average Cohen's κ of every participant pair in the relation annotation task (on the shared document). The average Cohen's κ is 0.7465. Details of instructions and interfaces are in Appendix D.1.

The statistics of the constructed datasets are shown in Table 1. The distributions of relation types are highly imbalanced, with a majority falling into either the *before* or *after* categories. We also evaluated the trained models on the MATRES test set (comprising 20 documents) and TBD test set (consisting of 9 documents), both of which are based on human annotations and processed into DOT using the methods previously described.

4.3 Model Setting

We use Flan-T5-base (Chung et al., 2022) (250M) as our backbone model. The model is trained for 10 epochs, with each document being augmented through 4 random permutations, followed by a further 3 epochs of training, during which the SPL loss is adopted without permutations. We use a learning rate of $2e-5$, along with a weight decay of 0.01. Batch size of 5 before SPL, and 3 during SPL because additional memory is required for sampling. We optimize the loss with the AdamW optimizer (Loshchilov and Hutter, 2019). We use the beam search (Graves, 2012) with a beam size of 5 and a maximum length of 2048 to sample results. We balanced the training steps in the compared methods to make sure they saw the same amount of training data. Experiments are conducted on a GPU node under an HPC cluster using 4 Nvidia A100 GPUs. The models are trained based on 3 random seeds (ChatGPT was tested for 3 times) and the metrics are the average values of them.

	NYT-test			NYT-human		
	P^E	R^E	F_1^E	P^E	R^E	F_1^E
Flan-T5-base	51.27	32.43	39.73	22.61	25.88	24.14
SAFF (w/o aug)	50.28	34.82	41.15	25.80	32.13	28.62
SAFF (w/o SPL)	**51.88**	36.64	42.95	**27.08**	34.91	30.50
SAFF	50.97	**39.96**	**44.80**	25.92	**40.21**	**31.52**

Table 2: Edge-based metrics on the NYT datasets

	NYT-test			NYT-human		
	P^N	R^N	F_1^N	P^N	R^N	F_1^N
Flan-T5-base	**75.52**	58.24	65.76	53.36	47.66	50.35
SAFF (w/o aug)	75.34	60.64	67.20	**54.86**	50.43	52.55
SAFF (w/o SPL)	75.43	62.36	68.27	54.14	51.59	52.84
SAFF	75.47	**65.16**	**69.95**	53.63	**54.51**	**54.06**

Table 3: Node-based metrics on the NYT datasets

4.4 Evaluation Metrics

Following the previous research (Madaan and Yang, 2021), we evaluate the results using the metrics of precision, recall, and F_1 score for both node set and edge set predictions. The primary metric is the edge F_1 because the quality of the node generation is also reflected in it.

4.5 Results

Table 2 and 3 show the evaluation results on NYT-test and NYT-human. `Flan-T5-base` was trained following the same setup as in Madaan and Yang (2021). `SAFF (w/o aug)` is the proposed framework without the augmentations of element order and with SPL. `SAFF (w/o SPL)` is the proposed framework without the use of Set Property Losses (SPL) and with the augmentations. As our SAFF framework with SPL requires additional training steps and the augmentations enlarge the training set, we keep the number of training steps balanced in the methods to exclude the influence of seeing different amounts of training data.

Comparing the results of `Flan-T5-base` with those of `SAFF (w/o SPL)`, it becomes evident that permutated augmentation improves edge F_1 by about 3% on NYT-test and 6% on NYT-human. While `SAFF (w/o aug)` shows the SPL alone can improve edge F_1 by about 1.5% on NYT-test and 4.5% on NYT-human. Furthermore, when both SPL and augmentation are incorporated, the graph generation performance is further increased (cf. `SAFF (w/o SPL)`/`SAFF (w/o aug)` and `SAFF`) We can also see that models utilizing SAFF have much higher edge recalls while their edge precision scores are either similar or occasionally even lower than those of other models. This suggests that the

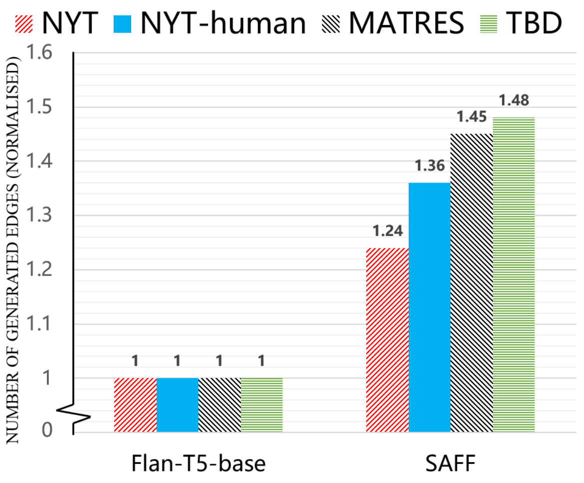

Figure 2: The comparison of generated edges between SAFF and vanilla `Flan-T5-base`. The y axis is normalised by dividing the number of edges generated by `Flan-T5-base` in the respective datasets.

	MATRES-test			TBD-test		
	P^E	R^E	F_1^E	P^E	R^E	F_1^E
ChatGPT	10.58	6.56	8.09	25.92	5.94	9.66
Flan-T5-base	13.06	7.16	9.25	23.26	4.59	7.67
SAFF	**18.05**	**14.31**	**15.96**	**37.53**	**11.04**	**17.05**

Table 4: Experiment results on human-annotated MATRES and TBD under the zero-shot setting.

performance improvement primarily comes from the generation of more edges. This observation is reinforced by the information presented in Figure 2, where models trained with SAFF can generate $24\% - 48\%$ more edges compared to the conventional text generation framework on these datasets. These additional edges play a pivotal role in the improvement of the edge F_1 since precision stays nearly the same.

It is worth mentioning that the NYT-human dataset has a different label distribution compared to the NYT dataset used for training, where its events and event temporal relations were produced by CAEVO. Notably, the frequency of *simultaneous* is significantly higher, accounting for 9.66%, in contrast to the 2.39% observed in the training set (see Appendix D.1 for more comprehensive analyses). Based on our observation, it appears that human annotators tend to apply a more lenient criterion for the *simultaneous* label whereas CAEVO enforces a stricter definition of this label.

Similar trends are also observed in Table 4, which were obtained through evaluation on MATRES and TBD. We used the models trained on the NYT training set to test on these datasets under zero-shot settings. It is worth noticing that SAFF exhibits superior zero-shot performance compared to GPT-2, as reported in (Madaan and Yang, 2021), despite our SAFF training using date four times smaller and GPT-2 (with 355M parameters) having 40% more parameters.

`ChatGPT` shows our best attempts to generate event temporal graphs with gpt-3.5-turbo API. We used two hops: (i) ask ChatGPT to generate events from the documents, (ii) ask ChatGPT to generate an event temporal graph based on the generated events and the documents. The results show that ChatGPT is outperformed by fine-tuned models, which is in line with the recent papers on exploring ChatGPT's ability on event understanding (Li et al., 2023; Chan et al., 2023; Gao et al., 2023). These facts confirm that event temporal graph generation cannot be solved solely with prompt engineering on ChatGPT. We provide the detailed inputs, outputs, and parameter settings in Appendix E.

4.6 Error Analysis

A major error type we found is that the model often fails to deduce temporal relationships that involve inference. This is due to the reliance of weak supervision signals provided by CAEVO, which primarily rely on syntactical rules. Consequently, this problem led to a lower edge F_1 on the human-annotated test set, as human annotators provided many temporal relations that were inferred through commonsense reasoning. Conversely, the model does not perceive a clear temporal sequence in the sentence:"<person A> won the gold medal in women's 1,500m. <person B> won the silver and <person C> won the bronze." However, human annotators can readily identify an obvious temporal order among "<person A> won", "<person B> won", and "<person C> won", as it aligns with the common knowledge that in a race, the first person who crossed the finish line won the gold, followed by the silver and the bronze winners.

5 Conclusion

This study proposes a framework for fine-tuning large language models to generate event temporal graphs directly from raw documents. We propose data augmentation and set property losses to mitigate the problem caused by conventional generation loss, promoting the generation of more edges by language models and, consequently, leading to improved performance.

Limitations

Due to the presence of noisy labels used in fine-tuning, a major limitation of the proposed method is the inclusion of many imaginary events, trivial events, and negative expressions of events. For example, CAEVO identified phrases like "<someone> did not **fire**" as an event. While "fire" serves as a predicate and the notion of "did not **fire**" can hold narrative significance, it may not be entirely suitable within the context of event temporal graphs. This is because it is not about the occurrence of an action or a change of state, but rather describes the absence of an event. Similarly, in some articles, there are descriptions of multiple potential future developments, such as "he might **buy** product A". Including such expressions as events might introduce confusion into the event temporal graph, as these represent possibilities rather than actual occurrences. This problem mainly arises from the behavior of the CAEVO method, which primarily focuses on identifying fine-grained predicates as events. The resolution to this problem lies in obtaining better-quality supervision signals which focus on salient events (i.e., those events with relatively higher occurrences that are important to the narrative).

Ethics Statement

The proposed method analyses the text provided and extracts relevant information from it. The algorithm cannot acquire information beyond the boundary of the given text. Thus, any associated risks stem solely from the data itself. This research only utilised publicly available data. As long as the data input to the model is collected according to the relevant data policies and guidelines, the proposed method does not introduce further risks.

References

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer.

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.

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.

Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. 2022. Pix2seq: A language modeling framework for object detection. In *International Conference on Learning Representations*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Emden R. Gansner. 2006. Drawing graphs with dot.

Jun Gao, Huan Zhao, Changlong Yu, and Ruifeng Xu. 2023. Exploring the feasibility of chatgpt for event extraction.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *ArXiv*, abs/1211.3711.

Rujun Han, I-Hung Hsu, Mu Yang, Aram Galstyan, Ralph Weischedel, and Nanyun Peng. 2019a. Deep structured neural network for event temporal relation extraction. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 666–106, Hong Kong, China. Association for Computational Linguistics.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019b. 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, Hong Kong, China. Association for Computational Linguistics.

Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Aman Madaan, Dheeraj Rajagopal, Niket Tandon, Yiming Yang, and Antoine Bosselut. 2022. Conditional set generation using seq2seq models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4874–4896, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Dheeraj Rajagopal, Peter Clark, Yiming Yang, and Eduard Hovy. 2021. Think about it! improving defeasible reasoning by first modeling the question scenario. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6291–6310, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Bill McDowell, Nathanael Chambers, Alexander Ororbia II, and David Reitter. 2017. Event ordering with a generalized model for sieve prediction ranking. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 843–853, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Igor Melnyk, Pierre Dognin, and Payel Das. 2022. Knowledge graph generation from text. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1610–1622, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019. An improved neural baseline for temporal relation extraction. 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 6203–6209, Hong Kong, China. Association for Computational Linguistics.

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

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

Kechen Qin, Cheng Li, Virgil Pavlu, and Javed Aslam. 2019. Adapting RNN sequence prediction model to multi-label set prediction. 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 3181–3190, Minneapolis, Minnesota. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Oliver Schutze, Xavier Esquivel, Adriana Lara, and Carlos A. Coello Coello. 2012. Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522.

Xingwei Tan, Gabriele Pergola, and Yulan He. 2021a. Extracting event temporal relations via hyperbolic geometry. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8065–8077, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xingwei Tan, Gabriele Pergola, and Yulan He. 2023. Event temporal relation extraction with Bayesian translational model. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1125–1138, Dubrovnik, Croatia. Association for Computational Linguistics.

Zeqi Tan, Yongliang Shen, Shuai Zhang, Weiming Lu, and Yueting Zhuang. 2021b. A sequence-to-set network for nested named entity recognition. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence, IJCAI-21*.

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.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2016. Order matters: Sequence to sequence for sets.

In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

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

Xinyu Wang, Lin Gui, and Yulan He. 2023. Document-level multi-event extraction with event proxy nodes and hausdorff distance minimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10118–10133, Toronto, Canada. Association for Computational Linguistics.

Haoyang Wen and Heng Ji. 2021. Utilizing relative event time to enhance event-event temporal relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10431–10437, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021. One2Set: Generating diverse keyphrases as a set. 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 4598–4608, Online. Association for Computational Linguistics.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2023. An autoregressive text-to-graph framework for joint entity and relation extraction. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*.

A Discussion of Alternative Approaches for Incorporating SPL

In our preliminary studies, we tested several way of incorporating SPL into the training process. We tried using the weighted average of SPL and the language modelling loss in every training step (the weight of SPL increases across the training process). However, the training becomes very slow due to the added decoding processes.

We also experimented with introducing randomness into the incorporation of the SPL loss in some of the training steps. In each training step, there was a 0.5 probability that the SPL loss was computed and backpropagated, while in other cases, only the LM loss was considered. This probability increased progressively with the epoch number. For example, during the initial epoch (epoch 1), the probability was set to 0, as we explained in Section 3.4, the model struggled to generate outputs in the correct format during the early stages. Subsequently, the probability was increased linearly, reaching 1 by the last epoch.

However, this approach proved to be ineffective and caused training instability. The loss value fluctuated between steps, leading to confusion for the model. It is worth noting that this was an initial experiment and significantly differed from the final version of the proposed method.

B Additional Error Analysis

We observed a type of error involving the model's incorrect prediction of long-distance temporal relationships. The model sometimes predicts a temporal relation between two events that are separated by more than ten sentences. This is unexpected, as the CAEVO model, which produces weak supervision signals, typically does not extract relations for events that are more than two sentences apart from each other. In essence, it primarily focuses on events within close proximity. Our observations suggest that human annotators also tend not to annotate temporal relations for events that are distant from each other, arguably because such relations are often implicit and can be challenging to track across large chunks of text.

C More Analysis about the Generation Results

Table A1 shows a preliminary experiment in which the augmented Flan-T5-base models were trained and tested on NYT-test before and after merging reciprocal relation types. The first row is the model trained with *before*, *after*, *includes*, *is_included*, and *simultaneous*. The second row is the model trained by merging *after* with *before*, *is_included* and *includes* by swapping the head and tail events. Both models are trained with 4 augmented instances for each original instance. The results show the model benefits from the simpler label set.

	P^N	R^N	F_1^N	P^E	R^E	F_1^E
With reciprocal relations	76.05	55.05	63.87	52.07	30.90	38.78
Merge reciprocal	75.48	61.96	**68.05**	52.03	36.34	**42.79**

Table A1: Comparison between model trained with reciprocal relations or merging reciprocal relations.

	before	*includes*	*simultaneous*
Target graph	93.13	4.63	2.24
Flan-T5-base	92.82	3.19	3.99
T5-base	92.48	3.94	3.59
SAFF (Flan-T5-base)	93.45	3.28	3.27
SAFF (T5-base)	93.00	3.65	3.35

Table A2: Generated graph temporal relation label distribution (in percentage).

Table A2 shows the relation type distribution generated by the models. The relation distributions are highly unbalanced.

Table A3 shows the average degree for the nodes in the generated graphs. The SAFF generate more complex graphs with higher node degree than the compared approaches.

We also investigate the effect of the order of the target edge set (Figure A4). The first row is Flan-T5-base fine-tuned with target edge set ordered based on random order in the documents. The second row is Flan-T5-base fine-tuned with target edge set ordered based on their appearance order in the documents. We could observe that the appearance order results in slightly better performance than random order. Each method was run on 5 different random seeds and trained for 50 epochs.

	average node degree
Vanilla Flan-T5-base	2.06
SAFF (w/o SPL)	2.16
SAFF	2.31

Table A3: The average node degree of the generated graphs on NYT.

	NYT-test			NYT-human		
	P^E	R^E	F_1^E	P^E	R^E	F_1^E
Random order	51.02	29.37	37.28	18.15	20.45	19.24
Appearance order	51.20	30.24	38.02	19.24	21.78	20.43

Table A4: Comparison based on different sequence orders on NYT-test and NYT-human.

D Annotation of the Test Set

D.1 Overview

We recruited crowd workers from Prolific[4] platform, which is a research-focused platform providing verified human workers. We recruited 24 participants in total (including pilot testing runs). In order to make sure the participants can understand and annotate the article efficiently, we require the participants to be native English speakers and have an education level higher than High school diploma/A-levels. We put 4 documents, which are randomly sampled from the same descriptor set as the training and testing of the selected NYT corpus, into each unit task. There is a shared document across all the tasks for the purpose of computing the inter-annotator agreement (IAA). In order to maximize the IAA, we asked 2 participants to first identify the event triggers in each unit task. After that, we merged the event annotations from the participants by taking the union of the spans (if there are overlapped spans, we take the longer span). Then, we asked another participant to annotate the event temporal relation based on the identified events. We also included the outputs from the CAEVO model to serve as examples, but we explicitly asked the participants to correct the annotations by adding, removing, or changing the CAEVO's annotations. In the end, we collected 22 documents as the human-annotated test set.

On the event identification, we compute IOU (Intersection over Union) as a measure of agreement between the annotators. Average across 7 tasks, the IOU between the event spans is 0.8986. For the relation annotations, we compute the average Cohen's κ of every participant pair in the relation annotation task (on the shared document). The average Cohen's κ is 0.7465.

D.2 Chosen Descriptors

Here are the chosen descriptors: "airlines and airplanes", "olympic games", "tennis", "united states international relations", "international relations", "civil war and guerrilla warfare", "track and field", "soccer", "bombs and explosives", "politics and government". We choose 2,000 documents from each descriptor. After preprocessing and filtering out some invalid documents, we have 18,263 documents in NYT-train, 1,000 documents in NYT-test, and 22 documents in NYT-human.

D.3 Instructions and Interface

We use a popular open-sourced annotation interface called Doccano. As shown in Figure A1, annotators can select text spans for events. To direct annotators to distinguish events that actually occurred and imaginary events, we also provide an "imaginary event" label type. We asked them to annotate the predicates that are about a negative expression of an action or just a hypothesis in the context as an imaginary event. Imaginary events are orthogonal to the real-world timeline and thus have limited meaning for understanding the narrative.

Figure A2 shows the interface for annotating the relation. On this page, annotators can select two existing event spans, and then select the relation type from "before", "includes", and "simultaneous".

Before the annotators came to the annotation platform, they went through a website where we put detailed descriptions and terminology definitions about the task. We also provided a video tutorial for using the annotation platform.

E ChatGPT prompting

We used the OpenAI API chat completion model *gpt-3.5-turbo-0613*. We used the "function call" method to ensure better parsing quality. The function call parameters are shown in Figure A5. The temperature is set to 0. The other parameters are set as default. We show the inputs and outputs of the multi-hop prompting in Table A5.

F GPT-4 Case Study

We show some test cases of where GPT-4 was prompted in this anonymous link [5]. The responses of GPT-4 essentially serve as summaries of the documents provided. The events it understood are quite broad, resembling abstracts of segments in the documents. This diverges from the NLP community's definition of event understanding, which typically pertains to the occurrence of specific actions. We would like to obtain more granular information within the event temporal graph.

[4] prolific.com

[5] Test cases on the TBD dataset

Another issue with the graphs generated by GPT-4 is that they tend to represent a linear sequence of items ordered by their appearances in the document. This ties back to the first issue, which relates to how GPT-4 comprehends events. It essentially generated a summary of the document, which, while not incorrect, does not align with the standard of event temporal graph extraction as defined in SemEval 2013 TempEval-3 (UzZaman et al., 2013).

Simply providing the definition of an event has not brought about a change in its behaviour [6]. While extensive prompt engineering might help, we believe that incorporating some supervision signals could still be necessary. Our framework could prove valuable for instruct-finetuning, aligning specific instructions with the event temporal graph generation task.

[6] Prompt with definition

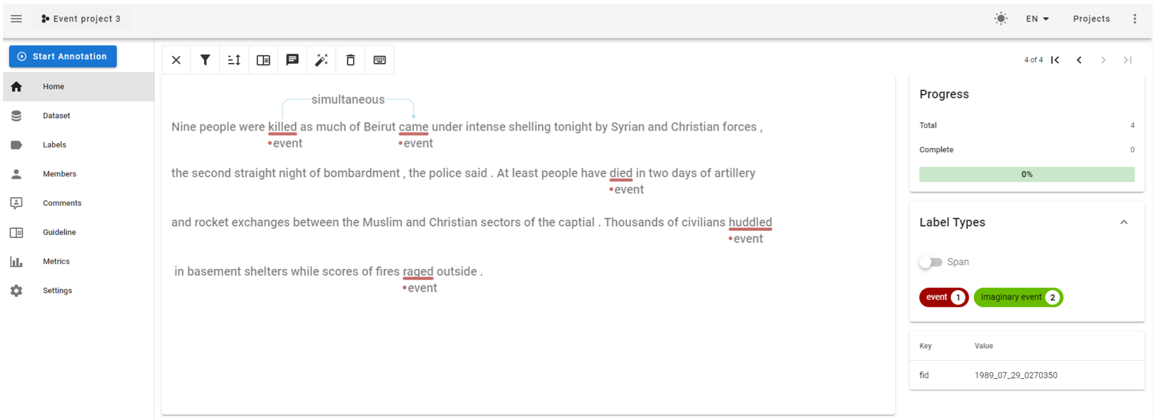

Figure A1: Annotation interface for event identification.

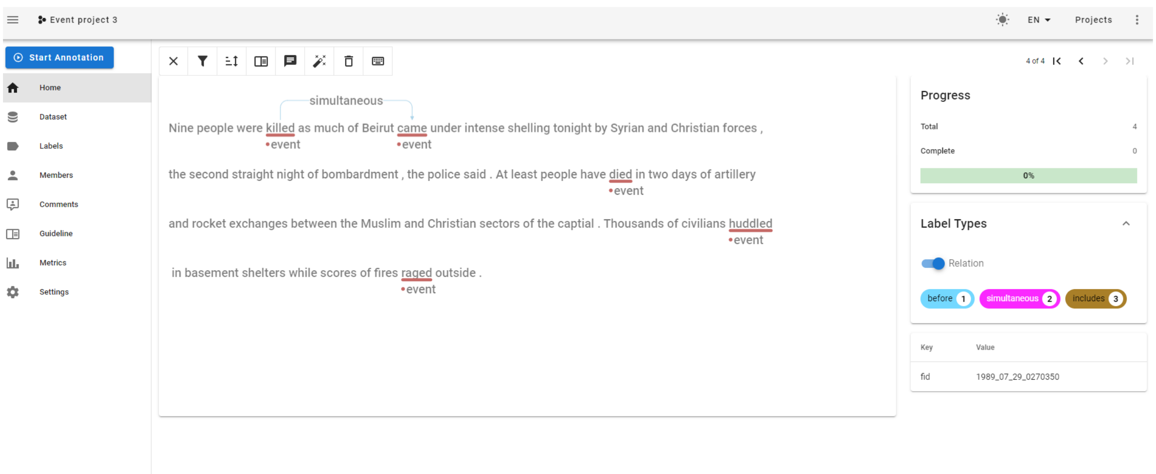

Figure A2: Annotation interface for event relation identification.

Participant Information Sheet

Invitation

You are invited to take part in a research study on event temporal relation understanding. The researchers in this project comes from [redacted] Please take the time to read the following information carefully. Please ask us if there is anything that is not clear or if you would like more information.

What will happen?

You will be asked to annotate a series of news articles. In each article, you will be asked to identify the temporal relation between the events. Specifically:

1. Given two event triggers, choose a temporal relation (definition provided below) type if the context indicates they are temporally related.

The interface of our annotation platform works best with a keyboard and mouse. Annotating with a touch screen can be very challenging.

Details of Annotation

Here is a detailed guideline for the annotation including images and videos as examples.

Time Commitment

You will be asked to annotate several news articles. The quantity are based on the length of each document. We estimate the task should take approximately an hour of your time.

Confidentiality

We won't collect any of your personal data as part of this study. We will only collect your Prolific ID for the purpose of assigning you an anonymous account on our annotation platform. Once your responses have been submitted, if you wish to withdraw the data you have provided, you will be able to do so by emailing [redacted] and quoting your Prolific ID. We will only retain a record of your Prolific ID for 5 years after the end of this study (I.e., up until Sep 2028). After that time, it will no longer be possible to identify your individual responses.

Participant rights

Your participation in this study is completely voluntary and choosing not to take part will not affect you or your rights in any way. You can also choose to withdraw your participation at any time, without giving a reason. You have the right to not answer or respond to any question that is asked of you. If you wish to withdraw from participating in the study, please close your browser.

Benefits and risks

There are no known benefits or risks beyond everyday life for you in this study. The study will be useful in understanding what makes a conversation interesting to those involved in it.

Figure A3: Disclaimers.

Guidelines

Task Overview

You will be asked to annotate a series of news articles. In each article, you will be asked to identify the events mentioned in text and the temporal relation between them. Specifically:

1. Given two event triggers, choose a temporal relation (definition provided below) type if the context indicates they are temporally related.

The interface of our annotation platform works best with a keyboard and mouse. Annotating with a touch screen can be very challenging.

Definition of the Terminologies

1. Event: An event is an action or a change of states. Events in text are notified by event triggers, which is the word that indicates the occurrence of the event. A trigger is usually a predicate, but it can also be a noun phrase. **In this task, the event triggers have been annotated.**
2. Temporal Relation: We are interested in the relative temporal relations between events. We have the following label types: *before*, *includes*, and *simultaneous*.
    - If event A ends before event B starts, then event A happened *before* event B.
    - If event A happened within the span of event B, then event B *includes* event A.
    - If event A and event B clearly don't include one or the other, and the time spans of them mostly overlap, then choose *simultaneous*.

    We provide some examples below, please check them carefully before the task. Note that you don't have to choose a relation for every event pair. **Only when the context clearly indicicates a temporal relation**, then choose a relation label. Otherwise, please skip them.

Annotation Tutorial

Please ignore the part about annotating the event triggers. In this task, you are only responsible for relations.

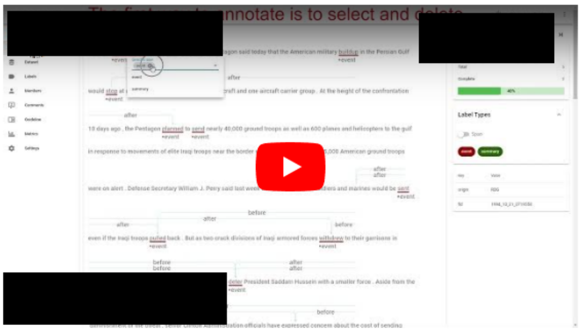

Figure A4: Guides.

```
FUNCTION_LIST = [
    {
        "name": "save_events",
        "description": "Store the extracted events in a list",
        "parameters": {
            "type": "object",
            "properties": {
                "event_list": {
                    "type": "string",
                    "description": "This is a list of event strings",
                }
            },
            "required": ["event_list"],
        },
    },
    {
        "name": "save_graph",
        "description": "Store the constructed graph in DOT language",
        "parameters": {
            "type": "object",
            "properties": {
                "graph": {
                    "type": "string",
                    "description": "The constructed graph in DOT language. \
                    This graph is a strict graph, in which every edge containing \
                    two event nodes, and a temporal relation label from \
                    [\"before\", \"includes\", \"simultaneous\"]. For example, \
                    \"strict graph  {\n\"The Organization asserted responsibility \
                   \" -- \"a United States Navy diver killed\"  [rel=before];\n}\"",
                }
            },
            "required": ["graph"],
        },
    }
]
```

Figure A5: Function list for OpenAI API call

Input	Output

First hop:
Instructions: Analyse the given document. Extract the main events. Store the events as a JSON list by calling the save$_{e}ventsfunction.$
Example input document: Governor Cuomo leaving his office in Albany yesterday to speak to reporters after he met with representatives of group pushing for higher ethical standards for public officials in the state.
Example output events: Governor Cuomo leaving his office, Governor Cuomo speak to reporters, Governor Cuomo met with representatives."
Document: President Clinton says he is committed to a possible strike against Iraq, despite objections from other countries. Steve Hurst has that. The US military buildup in the Persian gulf continues apace, more planes headed from the united states and senior officials say Iraq's president Saddam Hussein can expect punishing air strikes to continue well into the future if he doesn't stop building biological and chemical weapons. Saddam will know by our actions and our warning that we will be prepared to act again if we have evidence he is trying to rebuild his weapons of mass destruction capabilities. In a bit of television diplomacy, Iraq's deputy foreign minister responded from Baghdad in less than one hour, saying Washington would break international law by attacking without UN approval. The united states is not authorized to use force before going to the council. President Clinton, meantime, glossed over stern warnings from Moscow on Thursday that US air strikes against Iraq could do serious harm to relations with the Kremlin. We're trying to find a diplomatic solution. And I hope that, whatever happens today, that our relationships with Russia will continue to be productive and constructive and strong, because that's very important to the future of our peoples. One contrary view of the issue presented itself to the president as he arrived in Philadelphia later in the day. Nevertheless, the president said Washington would use force if diplomacy fails to force Saddam Hussein to back down.
	strict graph {
	"President Clinton's commitment to a possible strike against Iraq" -> "The US military buildup in the Persian Gulf" [label="before"];
	"The US military buildup in the Persian Gulf" -> "Senior officials' statement about punishing air strikes against Iraq" [label="before"];
	"Senior officials' statement about punishing air strikes against Iraq" -> "Iraq's deputy foreign minister's response from Baghdad" [label="before"];
	"President Clinton's attempt to find a diplomatic solution" -> "Contrary views expressed in Philadelphia" [label="before"];
	"Contrary views expressed in Philadelphia" -> "President Clinton's statement about using force if diplomacy fails" [label="before"];
	"The Russian foreign minister's suggestion to wait for UN Secretary General Kofi Annan's visit" -> "Secretary of State Madeleine Albright, Berger, and Defense Secretary William Cohen's plans to travel and explain military force necessity" [label="before"];
	}

The Russian foreign minister, meanwhile, sought to soften the harsh words of his military counterpart, saying on Friday that Russia now feels the US must hold off at least until UN secretary general Kofi Annan visits Baghdad in a last-ditch effort at diplomacy. Annan has no trip planned so far. Meanwhile, secretary of state Madeleine Albright, Berger and defense secretary William Cohen announced plans to travel to an unnamed city in the US heartland next week, to explain to the American people just why military force will be necessary if diplomacy fails. Steve Hurst, CNN, the state department.

Second hop:

For every possible pair of events, please infer the relative temporal order (e.g., before, includes, simultaneous) between them based on the context. Make a graph where the nodes are the events, and the edges represent the temporal relation between the events. Lastly, please output the graph using DOT graph description language.

Conversation link:

https://chat.openai.com/share/a7c8659d-4fbd-4cfa-b60c-b4066ac77ee9

First hop:
Instructions: Analyse the given document. Extract the main events. Store the events as a JSON list by calling the save_events function.
Document: For The World, this is Pamela McCall in the BBC news room in London. More heavy gunfire in the Serbian province of Kosovo. It's the second day of an offensive by the security forces against villages populated by ethnic Albanians. BBC correspondent Karyn Coleman reports from Kosovo. There were more accounts of people fleeing from the villages near where the offensive was taking place. They were too afraid to stay, fearing the forces may also move their operations further afield. By mid afternoon, official Serb sources were saying the operation was over, but that has not yet been confirmed from Belgrade, the capital of Serbia, which is where the whole attack is thought to have been planned. The ethnic Albanians here in Kosovo are considerably worried about the future and fear they are witnessing the beginnings of a much wider conflict. Karyn Coleman, BBC news, Pristina.
Second hop:
For every possible pair of events, please infer the relative temporal order (e.g., before, includes, simultaneous) between them based on the context. Make a graph where the nodes are the events, and the edges represent the temporal relation between the events. Lastly, please output the graph using DOT graph description language.
Conversation link:
`https://chat.openai.com/share/6bdeb237-247d-486f-91ba-146c3b4fd3b2`
	strict graph {
	"Heavy gunfire in the Serbian province of Kosovo." – "The second day of an offensive by the security forces against villages populated by ethnic Albanians" [rel=before];
	"The second day of an offensive by the security forces against villages populated by ethnic Albanians" – "People fleeing from villages near the offensive" [rel=before];
	"The second day of an offensive by the security forces against villages populated by ethnic Albanians" – "Official Serb sources stating the operation was over" [rel=before];
	"Official Serb sources stating the operation was over" – "Uncertainty regarding the operation's status in Belgrade" [rel=simultaneous];
	"Uncertainty regarding the operation's status in Belgrade" – "Ethnic Albanians in Kosovo expressing worry about the future and fear of a wider conflict" [rel=before];
	}

Table A5: Example of ChatGPT promoting on the TBD data.