Contrastively Pre-trained Event Embeddings with Schema-free LLM Annotations

Frank Mtumbuka, Steven Schockaert
Cardiff University, UK
{mbtumbukaf,schockaerts1}@cardiff.ac.uk

Abstract

Event extraction is a notoriously challenging problem, among others due to the scarcity of suitable training data. Moreover, event-centric knowledge bases are not available for most domains, making traditional distant supervision strategies difficult to implement. In this paper, we evaluate the potential of using LLM-generated annotations as an alternative distant supervision signal. Specifically, we create a synthetically labelled event extraction corpus, using an LLM to identify event triggers and arguments, and to provide corresponding free-text descriptions. We then pre-train event embedding models on this corpus using a contrastive loss, before fine-tuning them in the usual way. We empirically show the effectiveness of this approach.

Keywords: distant supervision, event extraction, synthetic data

1. Introduction

The problem of Event Extraction (EE) consists in uncovering the structure of the events that are mentioned in text. This problem is composed of several sub-tasks, including identifying the trigger (i.e. the word or phrase that refers to the event) and arguments, as well as predicting their types. Consider the following example (Wang et al., 2024):

> British losses were confined to a single man wounded by accident aboard "Crescent".

Using the MAVEN-ARG (Wang et al., 2024) annotation scheme, *accident* is the trigger and the event type is *Incident*. There are two arguments: *a single man wounded*, of type *Loss*, and *aboard "Crescent"*, of type *Location*. Annotating such examples is highly challenging, even for humans, which means that existing datasets tend to be small and often noisy. Moreover, existing datasets use different annotation schemas, with some being targeted to particular domains (Doddington et al., 2004) and others rather aiming for broad coverage (Wang et al., 2020). This means, in particular, that annotators need to carefully study detailed annotation guidelines to be capable of labelling examples. Peng et al. (2023) noted that LLMs with In-Context Learning (ICL) struggle with such specification-heavy tasks. Accordingly, the current state-of-the-art in event extraction is still based on supervised fine-tuning of smaller language models, such as those from the BERT (Devlin et al., 2019) family.

The fact that fine-tuned BERT models can outperform LLMs with ICL has been observed in other contexts as well (Edwards and Camacho-Collados, 2024; Bucher and Martini, 2024). Fine-tuned BERT models also clearly have the advantage of being significantly more efficient than LLMs. Thus, even if we can reasonably expect that LLMs will at some point outperform such models (either through the introduction of newer models or the discovery of more effective prompting strategies), smaller models are likely to remain important. However, the success of fine-tuned smaller models critically depends on the quality of the available training data, which is clearly a limiting factor in the case of event extraction. A common strategy for alleviating data scarcity is to rely on distant supervision. For instance, relation extraction systems are often pre-trained by leveraging knowledge graphs (Riedel et al., 2010; Elsahar et al., 2018; Huguet Cabot and Navigli, 2021). For event extraction, such approaches are harder to use due to the lack of comprehensive event-centric knowledge bases, and the difficulty of aligning such structured event knowledge with mentions in text. Although some approaches have been proposed (Reschke et al., 2014; Chen et al., 2017; Wang et al., 2022b, 2021), their effectiveness has been mostly limited specialized domains, where suitable knowledge bases are available.

In this paper, we analyze the potential of LLMs for providing us with a high-quality distant supervision signal for pre-training event extraction systems. Specifically, we use GPT-4o to annotate a large number of sentences, asking the model to identify event triggers and arguments, and to provide corresponding free-text descriptions. One core challenge, when using LLMs for event extraction, is that it is challenging to align LLM outputs to a specific schema (Srivastava et al., 2025). Accordingly, in our approach, we do not rely on any predefined schema. Instead, we use the generated free-text descriptions as part of a contrastive pre-training strategy, where we fine-tune a BERT encoder such that events with similar descriptions are encoded in a similar way. This pre-training process is illus-

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 7457–7478
11–16 May 2026.

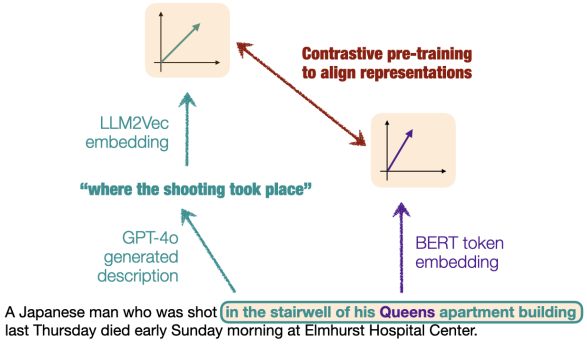

Figure 1: Schematic overview of the pre-training process. Event triggers and arguments are identified using GPT-4o, which also provides the corresponding descriptions. The LLM2Vec embeddings of these descriptions are aligned with BERT-generated token embeddings.

trated in Figure 1. After the pre-training phase, the encoder is fine-tuned on a standard training set. Intuitively, the pre-training step teaches the model to represent event types and argument roles, whereas fine-tuning is then mostly aimed at learning how to map these representations onto a given label set.[1]

Our main contributions are as follows:

- We create an event extraction corpus, consisting of 150k sentences that were annotated with GPT-4o. We also provide annotations that were generated by GPT-4o-mini for a subset of 100k sentences.
- We demonstrate the effectiveness of pre-training event encoders on this corpus, obtaining consistent and substantial gains on three standard benchmarks: MAVEN-ARG (Wang et al., 2024), MAVEN (Wang et al., 2020) and ACE-2005 (Christopher Walker, Stephanie Strassel, Julie Medero, Kazuaki Maeda, 2006).
- We show that relying on GPT-4o-mini is largely ineffective, demonstrating the need for high-quality annotations.

2. Related Work

EE has traditionally used supervised approaches, fine-tuning encoders of the BERT family or generative models such as T5 (Raffel et al., 2020). More recently, various works have explored the potential of Large Language Models (LLMs) in this context, especially for zero-shot and few-shot settings.

[1]Our code and datasets are available at: https://github.com/fmtumbuka/restore-event-extraction.

Supervised Event Extraction Event extraction methods have traditionally relied on supervised learning, which requires fine-grained annotations at the token and entity levels. Encoder-only models of the BERT family remain one of the most popular choices for training event extraction models. For instance, some models formulate EE as a classification problem, by first identifying candidate mentions and then assigning labels to these mentions from a predefined schema (Wang et al., 2021). Several approaches take into account the interdependencies that exist between the predictions of event types and argument roles, and in some cases also entity types (Nguyen and Nguyen, 2018; Wadden et al., 2019; Lin et al., 2020). Other approaches formulate EE as a sequence labelling problem (Chen et al., 2018). Yet another possibility is to cast EE as a question answering or span extraction problem (Du and Cardie, 2020; Li et al., 2020; Liu et al., 2020; Ma et al., 2022). Note that this latter formulation opens the door to using generative language models, including encoder-decoder models such as T5 (Raffel et al., 2020) and decoder-only models, such as those from the GPT family (Radford et al., 2019). Generative models also enable a formulation of EE as a sequence-to-sequence problem (Lu et al., 2021; Paolini et al., 2021; Li et al., 2021; Hsu et al., 2022; Liu et al., 2022).

Distant Supervision Distant supervision strategies for *relation extraction* heavily rely on the availability of accurate entity-linking methods, which can align mentions of entities in text with their corresponding entries in a knowledge base, and on the availability of large-scale knowledge bases. Distant supervision for *event extraction* is considerably more challenging, due to a lack of comprehensive event databases, in most domains, and the lack of readily available tools for aligning mentions of events in text with structured sources. Nonetheless, several distant supervision strategies for event extraction have already been explored. Some authors have focused on domain-specific settings, such as airplane crash events (Reschke et al., 2014) the finance domain (Yang et al., 2018) and the biomedical domain (Wang et al., 2022b), where specialized knowledge sources can be leveraged. Chen et al. (2017) and Zeng et al. (2018) use event knowledge from Freebase, proposing heuristic methods to identify the key arguments of each event. An important challenge is to identify the event triggers, as these cannot be readily linked to the knowledge base. Chen et al. (2017) use a frequency-based heuristic to identify trigger verbs, and then leverage FrameNet to find nominal references and filter noisy annotations. Yang et al. (2018) constructed a dictionary of trigger words for each event type, which is only feasible in restricted domains. Araki

and Mitamura (2018) proposed to identify event triggers by using word-sense disambiguation and applying a set of manually constructed rules. They also leveraged off-the-shelf strategies for linking event mentions to their Wikipedia article, and proposed methods for deciding if a given Wikipedia article is about an event. Wang et al. (2019) suggested a data augmentation strategy based on the idea that whenever a trigger word from a known event instance appears in another sentence, this other sentence may express an event of the same type. Adversarial training was used to mitigate the noisy nature of this heuristic. Finally, Wang et al. (2021) used contrastive learning to align an event extraction model with Abstract Meaning Representation parses. Compared to this latter strategy, our approach is novel in its use of schema-free LLM annotations, whereas AMR parses are tied to predefined schemas. Moreover, AMR parsing is notoriously challenging and thus introduces noise.

LLMs for Event Extraction Applying LLMs to Information Extraction (IE) has been a topic of active research in recent years. LLMs have achieved strong results in zero-shot and few-shot IE tasks (Wei et al., 2023), but for problems where sufficient training examples are available, their usefulness remains unclear. For example, Sharif et al. (2024) analyzed ChatGPT's performance on event detection in the health domain and found it inferior to supervised baselines. Similarly, Zhang et al. (2024) demonstrated that LLMs require additional guidance, such as hierarchical modeling, to improve their efficacy in extracting fine-grained arguments. One challenge relates to the problem of hallucination: LLMs often produce plausible but incorrect outputs, especially when confronted with complex, long-tail or domain-specific contexts (Ji et al., 2023; Li et al., 2023; Gao et al., 2023). Crafting effective prompts for EE also continues to be a challenge, as LLMs are highly sensitive to how instructions are framed (Gao et al., 2023). Finally, compared to supervised systems, LLMs are found to perform poorly in tasks requiring precise token-level predictions (Han et al., 2023; Zhang et al., 2024; Huang et al., 2024). As an alternative to using LLMs directly, Meng et al. (2024) leveraged LLMs for sentence-level data augmentation. Some authors have also resorted to fine-tuning (smaller) LLMs to address some of the aforementioned shortcomings. For instance, Sainz et al. (2024) fine-tune an LLM to make it comply with (previously unseen) annotation guidelines, and show how this improves zero-shot IE. Several authors have also fine-tuned LLMs to improve their performance across a range of different IE tasks, by using a unified output format (Lu et al., 2022; Lou et al., 2023; Wang et al., 2023; Li et al., 2024). These models perform strongly in low-resource settings, but improvements for the fully supervised case are less consistent.

Input Sentence: On Saturday, the French asked the U.N. Security Council to authorize the creation of a safe haven in southwestern Rwanda.
Trigger span: asked **Description**: The French requested authorization from the U.N. Security Council for a safe haven in Rwanda.
Argument span: the French **Role**: proposer **Description**: The entity making the request
Argument span: the U.N. Security Council **Role**: addressee **Description**: The entity being asked for authorization
Argument span: to authorize the creation of a safe haven **Role**: action **Description**: The action requested by the French
Argument span: in southwestern Rwanda **Role**: location **Description**: The proposed location for the safe haven
Argument span: On Saturday **Role**: time **Description**: When the request was made

Table 1: Example annotation from GPT-4o.

3. Methodology

We follow a standard distant supervision pipeline. In particular, we first pre-train a generic event encoder on a large number of sentences that were automatically annotated, in our case, using an LLM. This enables the model to develop a general understanding of events and their arguments. Subsequently, we fine-tune the encoder using a traditional training dataset, to adapt it to a given domain, and map its representations onto the labels from the considered schema.

3.1. Pre-training Step

Obtaining LLM Descriptions To construct the pre-training dataset, we annotate sentences from an unlabelled corpus using an LLM. For the experiments in this paper, we select these sentences randomly from the Gigaword news corpus (David Graff, Junbo Kong, Ke Chen, Kazuaki Maeda, 2005) and use GPT-4o[2] as the LLM. Given a sentence, we prompt the LLM to list all events and provide the following details for each event: the event trigger (and

[2] https://openai.com/index/gpt-4o-system-card/

its position), a free-text description of the event, and a list of arguments. For each argument, the LLM is furthermore prompted to provide the corresponding text span (and its position), a label for the role, and a free-text description of the role. Table 1 shows the output that was generated for a sample sentence. Note that in this case, the model identified a single event, but in general multiple events may be extracted. The prompt uses a structured JSON format and includes an in-context example, to ensure a consistent output format.

Pre-training Objective In the generated pre-training corpus, we have access to two different representations of event triggers and arguments: as a span from the original sentence s and as an LLM-generated description d. We use a (frozen) embedding model ψ to obtain a vector $\psi(d)$ for each LLM-generated description d. Using a BERT encoder, we furthermore obtain a vector encoding $\phi(x_i; s)$ of each token x_i in the span. The main idea of our approach is to fine-tune this BERT encoder using a contrastive loss (van den Oord et al., 2018), such that the representations $\psi(d)$ and $\phi(x_i; s)$ are aligned. In other words, we want the vectors $\psi(d)$ and $\phi(x_i; s)$ to be similar if d is the description of the span containing x_i, and dissimilar otherwise.

Let us consider a training batch with n annotated spans (i.e. event triggers or arguments). Let us write s_i for the sentence corresponding to training example i ($i \in \{1, ..., n\}$), x_i for a token from some annotated span in that sentence, and d_i for the description of that span. We write $\phi(x_i; s_i)$ for the encoding of the span x_i and $\psi(d_i)$ for the encoding of the description d_i. We then use the following InfoNCE loss as our pre-training objective:

$$-\frac{1}{|B|}\sum_{i=1}^{n}\log\frac{\exp\left(\frac{\cos(\phi(x_i;s_i),\psi(d_i))}{\tau}\right)}{\sum_{j=1}^{n}\exp\left(\frac{\cos(\phi(x_i;s_i),\psi(d_j))}{\tau}\right)}$$

where τ is a temperature parameter. For spans that consist of multiple tokens, we treat each token in the span as an independent positive example for contrastive learning. When pre-training the span encoder, we add the Masked Language Modeling (MLM) objective to prevent catastrophic forgetting, following standard practice (Baldini Soares et al., 2019). We mask 15% of the tokens from the trigger/argument spans. For event argument spans, we concatenate the label and the description provided by the LLM (e.g. "*time: when the request was made*", for the last example from Table 1). For event trigger spans, we simply use the description, as we found GPT-4o less consistent in providing high-quality labels for triggers.

3.2. Fine-tuning Step

The pre-training step encourages the model to learn informative span embeddings. However, in downstream tasks, the spans are typically not provided, so we need a strategy for identifying them. Moreover, we also need to learn a classification head, to map the embeddings to the labels from a given schema. We may also need to fine-tune the encoder itself, to adapt it to a given domain or genre. To accomplish these three objectives, we fine-tune the pre-trained encoder ϕ on a sequence labeling task, where each token from the input sentence is assigned a label. We consider the following labels:

- For each event type t, we have a label TRIGGER-t to indicate tokens that belong to the trigger of an event of type t.
- For each argument role r, we have a label ARG-r to indicate tokens that belong to an event argument with role r.
- We have the label OTHER to denote tokens that neither belong to an event trigger nor to an argument.

We use a standard neural sequence labeling model, where the output of the encoder for each token is fed to a linear layer, which predicts logits for each of the labels. These logits are then processed by a Conditional Random Field (CRF) layer (Lafferty et al., 2001), which predicts the final probability of each label. The purpose of this CRF layer is essentially to encourage neighbouring tokens to receive the same label, such that the model is more likely to predict coherent spans. The model is trained using categorical cross-entropy.

4. Experiments

We now empirically analyze the success of the proposed pre-training strategy on a number of standard event extraction benchmarks. We will refer to our proposed approach as COPELLA (Contrastive Pre-trained Event extraction with LLM Annotations).

4.1. Experimental Setup

The text span encoder ϕ is initialized with one of the following pre-trained language models: `bert-base-uncased`[3], `ModernBERT-base`[4], `bert-large-uncased`[5] and `roberta-large`[6]. For

[3] https://huggingface.co/google-bert/bert-base-uncased

[4] https://huggingface.co/answerdotai/ModernBERT-base

[5] https://huggingface.co/google-bert/bert-large-uncased

[6] https://huggingface.co/FacebookAI/roberta-large

		P	R	F1	EM
BERT-base + CRF†		31.7	31.4	30.9	27.0
CLEVE (RoBERTa-large)†		22.1	22.1	22.1	22.0
EEQA (BERT-base)†		21.4	19.5	19.6	15.8
Text2Event (T5-large)†		12.9	12.9	12.7	11.3
PAIE (BART-large)†		37.2	36.2	35.6	30.3
COPELLA-zero	BB	30.9	30.1	30.5	28.7
	BL	31.4	31.2	31.3	29.3
	MB	31.2	30.5	30.8	29.1
	RL	31.6	31.3	31.4	30.1
COPELLA 25k	BB	36.1	35.4	35.7	33.2
	BL	38.5	38.2	38.3	34.4
	MB	36.8	36.2	36.5	33.6
	RL	40.5	39.5	40.0	35.7
COPELLA 50k	BB	37.6	37.4	37.5	34.3
	BL	39.9	40.3	40.1	34.8
	MB	38.2	38.2	38.2	34.6
	RL	41.0	40.1	40.5	37.7
COPELLA 100k	BB	41.5	38.1	39.7	37.0
	BL	42.0	40.9	41.4	39.0
	MB	41.9	38.9	40.3	37.8
	RL	43.0	41.9	42.4	39.3
COPELLA 150k	BB	41.9	38.7	40.2	38.1
	BL	42.6	41.7	42.1	39.9
	MB	42.3	39.5	40.9	38.6
	RL	**43.4**	**42.6**	**43.0**	**40.0**

Table 2: Results on MAVEN-ARG with BERT-base (BB), BERT-large (BL), ModernBERT-base (MB) and RoBERTa-large (RL). Results with † were taken from Wang et al. (2024).

the description encoder ψ, we use a frozen LLM2Vec (BehnamGhader et al., 2024) model[7], followed by a learned linear layer to map the embeddings to the dimensionality of the encoder ϕ. We report results for pre-training datasets of up to 150k sentences.

Benchmarks We evaluate our models on standard event extraction benchmarks. **MAVEN-ARG** (Wang et al., 2024) covers 98,591 annotated events and 290,613 arguments, using a fine-grained schema with 162 event types and 612 argument roles. **MAVEN** (Wang et al., 2020) only focuses on event triggers, providing annotations for 118,732 mentions, also using a fine-grained schema with 168 event types. **ACE 2005** (Christopher Walker, Stephanie Strassel, Julie Medero, Kazuaki Maeda, 2006) covers 5349 event mentions, 33 event types and 36 argument roles.

Evaluation Metrics Event extraction is sometimes evaluated as a token-level classification task.

[7] https://huggingface.co/McGill-NLP/LLM2Vec-Meta-Llama-3-8B-Instruct-mntp

		P	R	F1
BiLSTM + CRF†		63.4	64.8	64.1
BERT + CRF†		65.0	70.9	67.8
MOGANED†		63.4	64.1	63.8
CLEVE (RoBERTA-large)		64.9	72.6	68.5
UniST (RoBERTa-base)		66.7	69.9	68.3
UniST (RoBERTa-large)		66.5	69.7	68.1
Query-and-extract		-	-	68.8
COPELLA-zero	BB	67.5	72.1	69.7
	BL	66.3	72.3	69.2
	MB	67.9	71.4	69.6
	RL	67.0	73.1	69.9
COPELLA 25k	BB	68.4	74.6	71.4
	BL	68.9	74.6	71.6
	MB	68.7	74.3	71.4
	RL	69.5	75.4	72.3
COPELLA 50k	BB	68.8	74.8	71.7
	BL	69.5	75.4	72.3
	MB	69.3	74.7	71.9
	RL	70.2	75.6	72.8
COPELLA 100k	BB	72.0	77.2	74.5
	BL	71.1	77.0	73.9
	MB	72.0	77.3	74.6
	RL	72.1	77.9	74.9
COPELLA 150k	BB	72.0	77.2	74.5
	BL	71.3	77.4	74.2
	MB	72.2	77.8	74.9
	RL	**73.0**	**78.8**	**75.8**

Table 3: Results for MAVEN with BERT-base (BB), BERT-large (BL), ModernBERT-base (MB) and RoBERTa-large (RL). Results with † were taken from Wang et al. (2020), other baseline results from the original papers.

Models are then evaluated in terms of micro precision, recall and F1 score. Following the literature, we evaluate MAVEN and MAVEN-ARG in this way. Wang et al. (2024) also reported Exact Match (EM). This metric provides a stricter evaluation by calculating the percentage of spans that match the ground truth exactly, both in terms of span boundaries and the predicted label. We will also report this metric for MAVEN-ARG. For ACE 2005, it is customary to report results for trigger and argument classification separately. These metrics again require that both the span boundary and predicted type match the ground truth exactly.

Baselines As our main baseline, we use a variant of our method which does not use the pre-training corpus. Instead, it uses the benchmark-specific training set both for the contrastive pre-training step and for subsequent fine-tuning. We will refer to this baseline as COPELLA-zero. To provide additional context, we also compare with published results for state-of-the-art models, where available. We

		Trigger-C	Arg-C
EEQA (BERT-base)$^{\diamond}$		72.4	53.3
Text2Event (T5-base)$^{\diamond}$		69.2	49.8
Text2Event (T5-large)$^{\diamond}$		71.9	53.8
DeepStruct (GLM-10B)†		69.8	56.2
OneIE (BERT-base)*		74.7	56.8
UIE (T5-large)$^{+}$		73.4	54.8
TAGPRIME (BERT-large)$^{\circ}$		74.6	58.3
DEGREE (BART-large)$^{\square}$		73.3	55.8
AMR-IE (RoBERTa-large)$^{\bullet}$		75.0	58.6
COPELLA-zero	BB	70.5	53.9
	BL	69.7	50.8
	MB	69.7	52.3
	RL	69.8	52.0
COPELLA 25k	BB	71.0	56.4
	BL	70.6	51.7
	MB	71.2	55.5
	RL	70.3	53.2
COPELLA 50k	BB	72.1	58.8
	BL	71.9	59.0
	MB	72.1	58.6
	RL	74.3	59.7
COPELLA 100k	BB	73.2	59.2
	BL	73.4	59.6
	MB	73.1	59.3
	RL	75.4	60.8
COPELLA 150k	BB	73.4	59.7
	BL	73.4	59.5
	MB	73.5	59.5
	RL	**75.8**	**61.2**

Table 4: Results for ACE 2005 with BERT-base (BB), BERT-large (BL), ModernBERT-base (MB) and RoBERTa-large (RL). Results with † were taken from Wang et al. (2022a), other baseline results from the original papers.

focus on the fully supervised setting. In particular, we do not consider recent LLM-based methods, which typically focus on zero-shot or few-shot event extraction and as a result significantly underperform the considered baselines (Wang et al., 2024).

4.2. Results

Table 2 summarizes the main results for MAVEN-ARG, where we evaluate how the performance of our method changes when the size of the pre-training corpus is increased from 25k to 150k GPT-4o annotated sentences. The results clearly show the effectiveness of the pre-training strategy. Using 150k annotated sentences, for instance, the state-of-the-art increases from 30.3 to 40.0 in terms of exact match, and from 36.7 to 43.0 in terms of span-based F1. We can clearly see the effect of increasing the amount of pre-training data, with consistent gains when going from 25k to 50k, 100k and 150k, across all four of the tested language models. The

		P	R	F1	EM
COPELLA-zero	BB	30.9	30.1	30.5	28.7
	BL	31.4	31.2	31.3	29.3
	MB	31.2	30.5	30.8	29.1
	RL	31.6	31.3	31.4	30.1
Trig-Desc	BB	31.8	31.2	31.5	29.1
	BL	34.0	33.7	33.8	30.2
	MB	32.5	32.0	32.2	29.5
	RL	35.9	35.1	35.5	31.5
Arg-Desc	BB	33.2	32.6	32.9	30.6
	BL	35.4	35.1	35.2	31.7
	MB	33.8	33.3	33.5	30.9
	RL	37.3	36.4	36.8	32.9
Arg-Joint	BB	34.7	34.0	34.4	31.9
	BL	37.0	36.7	36.9	33.0
	MB	35.3	34.8	35.1	32.2
	RL	38.9	37.9	38.4	34.3
Combined	BB	36.1	35.4	35.7	33.2
	BL	38.5	38.2	38.3	34.4
	MB	36.8	36.2	36.5	33.6
	RL	**40.5**	**39.5**	**40.0**	**35.7**

Table 5: Results for different pre-training strategies on MAVEN-ARG with BERT-base (BB), BERT-large (BL), ModernBERT-base (MB) and RoBERTa-large (RL), using 25K GPT-4o annotated sentences.

improvements are especially remarkable given the overall simplicity of our model, e.g. compared to sophisticated prompt-tuning based methods such as PAIE (Ma et al., 2022). It is also worth noting that even with BERT-base as the baseline, our model outperforms the current state-of-the-art.

Table 3 shows the results for MAVEN. We can again see clear improvements, with consistent gains as the number of sentences is increased, and a substantial improvement over the current state-of-the-art. Note that the *BERT-base + CFR* baseline is similar to *COPELLA-zero* (with BERT-base), except that for the latter model we contrastively pre-train the encoder on the training data. As can be seen, this pre-training step leads to improved results on this benchmark. As in the case of MAVEN-ARG, RoBERTa-large emerges as the best-performing language model. Interestingly, the difference with the other models is most pronounced for 150k sentences. While the performance of BERT-base saturates after 100k sentences, the performance of RoBERTa-large continues to improve.

Table 4 shows the results for ACE 2005. We can again see a clear benefit from our pre-training strategy, although the gains are overall smaller. This can be explained by the fact that the smaller number of labels in ACE 2005 can make prompting based methods more effective. Nonetheless, our model outperforms the existing methods when enough pre-training sentences are used: 100k for Trigger-

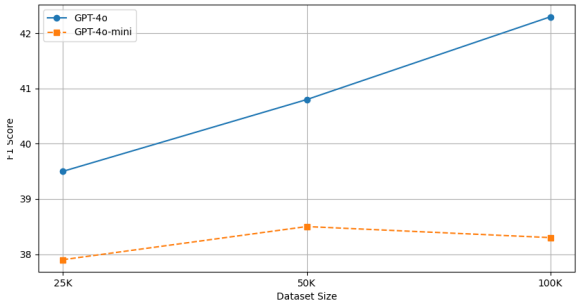

Figure 2: Comparing the performance of RoBERTa-large (RL) pre-trained on GPT-4o and GPT-4o-mini annotations across on the MAVEN-ARG dataset.

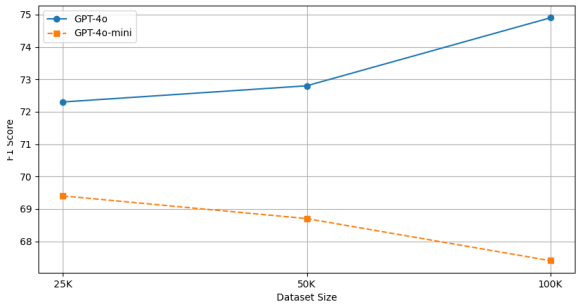

Figure 3: Comparing the performance of RoBERTa-large (RL) pre-trained on GPT-4o and GPT-4o-mini annotations across dataset sizes (25K, 50K, 100K) on the MAVEN dataset.

C and 50k for Arg-C (for RoBERTa-large).

4.3. Analysis

Ablation Analysis We consider four variants of the pre-training strategy, which differ in the kinds of spans and descriptions that are used:

Trigger-Description Alignment (Trig-Desc) only applies contrastive learning to the trigger spans, aligning the embeddings of the trigger tokens with the trigger description.

Argument-Description Alignment (Arg-Desc) only applies contrastive learning to the argument spans, aligning the embeddings of the argument tokens with argument description.

Argument-Joint Alignment (Arg-Joint) is similar to *Arg-Desc*, except that a concatenation of the argument role label (as predicted by GPT-4o) and the argument description is used (using phrases of the form "role: description").

Combined sums the losses of *Trig-Desc* and *Arg-Joint*, thus leveraging the annotations of both event triggers and arguments. This is the variant that was used for our main experiments.

In Table 5, we compare the different variants. For this experiment, we used 25k GPT-4o annotated sentences for pre-training and we evaluate on MAVEN-ARG. As can be seen, each variant improves over the *COPELLA-zero* baseline. Comparing *Arg-Desc* and *Arg-Joint*, we find that including the argument role label has a positive effect. As expected, *Combined* achieves the best results.

We now compare two different strategies for dealing with cases where the span x_i consists of multiple tokens:

- **Individual Token Contribution**: We treat each token in the span x_i as an independent positive example for contrastive learning. This is the strategy that we used in the main experiments.
- **Average Token Representations**: We define $\phi(x_i; s_i)$ to be the mean of the contextualized representations of the tokens in the span x_i.

These two strategies have complementary strengths. Averaging is efficient and offers a compact representation, making it suitable for high-level tasks such as event classification. Conversely, treating each token as an individual example preserves fine-grained distinctions, aligning better with token-level tasks such as trigger or argument identification, but at a higher computational cost.

We also compare a number of alternatives for the description encoder ψ. In the main experiments, we used a frozen LLM2Vec model. We now also experiment with (i) using the contextualised representation of the [CLS] token of a pre-trained RoBERTa model and (ii) doing the same but starting from a pre-trained SBERT (Reimers and Gurevych, 2019) model[8]. Since these are smaller models, for these alternative choices, we fine-tune the description encoder ψ jointly with the span encoder ϕ.

Table 6 presents an analysis of the two strategies for dealing with multi-token spans (*individual* and *average*) and the three considered description encoders. We consistently find that the "individual token contribution" strategy outperforms "average token representation". When it comes to the description encoders, the differences are less clear, with each of the three encoders achieving the best results in some configurations, although the best overall results are obtained with LLM2Vec. Among the four considered backbones, RoBERTa-large consistently achieves the best results.

Comparison of LLMs In our main experiments, we relied on annotations provided by GPT-4o. The total cost of collecting this data was around $150.

[8] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

			CLS				SBERT				LLM2Vec			
			P	R	F1	EM	P	R	F1	EM	P	R	F1	EM
25k	Average	BB	34.7	34.3	34.5	31.4	34.9	34.5	34.7	31.8	34.6	35.5	35.0	31.1
		BL	37.1	36.2	36.6	32.9	37.1	36.9	37.0	33.3	37.5	36.8	37.1	33.4
		MB	35.4	34.9	35.1	32.0	35.5	35.2	35.3	32.3	35.4	35.9	35.6	31.7
		RL	38.5	37.6	38.0	33.9	39.8	37.7	38.7	35.4	39.3	37.9	38.6	34.7
	Individual	BB	35.7	35.2	35.4	33.3	35.7	35.4	35.5	33.4	36.1	35.4	35.7	33.2
		BL	39.6	38.5	39.0	34.4	39.6	38.3	38.9	34.2	38.5	38.2	38.3	34.4
		MB	36.6	36.0	36.3	33.7	36.7	36.3	36.5	33.7	36.8	36.2	36.5	33.6
		RL	39.8	39.2	39.5	34.6	39.9	39.5	39.7	35.3	40.5	39.5	40.0	35.7
50k	Average	BB	35.6	35.8	35.7	32.7	36.2	35.6	35.9	32.9	36.1	34.9	35.5	32.4
		BL	39.1	37.1	38.1	33.5	38.7	38.4	38.5	33.9	38.5	37.9	38.2	34.9
		MB	36.5	36.2	36.3	33.0	36.8	36.3	36.5	33.4	36.7	35.6	36.1	33.2
		RL	39.7	39.2	39.4	35.1	40.1	39.3	39.7	35.6	40.0	38.6	39.3	36.1
	Individual	BB	37.2	37.0	37.1	34.2	37.9	37.4	37.6	34.4	37.6	37.4	37.5	34.3
		BL	40.4	40.0	40.2	34.9	41.0	40.1	40.5	35.5	39.9	40.3	40.1	34.8
		MB	39.7	38.4	39.0	36.0	38.8	38.1	38.4	34.8	38.2	38.2	38.2	34.6
		RL	40.9	40.8	40.8	35.9	41.5	41.0	41.2	36.7	41.0	40.1	40.5	37.7
100k	Average	BB	39.1	37.7	38.4	35.4	39.5	38.2	38.8	35.8	40.2	38.5	39.3	34.6
		BL	41.5	40.3	40.9	37.2	41.7	40.3	41.0	37.0	41.2	39.7	40.4	37.9
		MB	38.1	37.8	37.9	34.5	40.1	38.8	39.4	36.2	40.5	39.0	39.7	35.5
		RL	42.0	40.9	41.4	37.6	42.3	41.1	41.7	38.5	41.4	40.4	40.9	37.8
	Individual	BB	40.0	38.4	39.2	36.9	40.7	39.3	40.0	37.2	41.5	38.1	39.7	37.2
		BL	41.3	40.8	41.0	37.3	42.0	41.3	41.6	37.8	42.0	40.9	41.4	39.0
		MB	40.5	39.1	39.8	37.1	41.1	39.9	40.5	37.5	41.9	38.9	40.3	37.8
		RL	42.9	41.8	42.3	38.6	42.5	41.8	42.1	39.2	43.0	41.9	42.4	39.3
150k	Average	BB	40.0	38.0	39.0	35.7	40.3	38.6	39.4	36.4	41.0	39.1	40.0	35.2
		BL	41.8	41.4	41.6	38.2	41.8	40.9	41.3	38.2	41.5	40.5	41.0	38.6
		MB	40.6	38.8	39.7	36.5	40.7	39.3	40.0	37.0	41.4	39.5	40.4	36.1
		RL	42.6	41.5	42.0	38.4	42.9	41.5	42.2	39.0	42.3	41.1	41.7	38.2
	Individual	BB	41.0	38.8	39.9	37.4	41.4	39.4	40.4	37.9	41.9	38.7	40.2	38.1
		BL	42.4	41.4	41.9	38.3	42.7	41.8	42.2	38.7	42.6	41.7	42.1	39.9
		MB	41.5	39.6	40.5	37.7	41.8	40.0	40.9	38.3	42.3	39.5	40.9	38.6
		RL	43.9	42.8	43.3	39.0	43.5	42.3	42.9	39.2	43.4	42.6	43.0	40.0

Table 6: Performance Comparison Across Different Description Embedding Strategies During Pretraining, for different variants of COPELLA on MAVEN-ARG, using the *Combined* pre-training strategy.

Collecting datasets that are one or two orders of magnitude larger would likely give us significant further performance gains, but the associated cost would be prohibitive. To analyze whether cheaper models would enable us to achieve comparable results, we therefore also annotated the first 100K sentences using GPT-4o-mini. Figure 2 compares the performance of the resulting models on MAVEN-ARG, while Figure 3 shows the same for MAVEN. While the F1 score gap between GPT-4o and GPT-4o-mini annotations is small at 25K, it widens significantly at 50K and 100K. For GPT-4o, the performance continues to increase as more sentences are added, whereas GPT-4o-mini plateaus for MAVEN-ARG. In the case of MAVEN, we even find that performance becomes progressively worse as more GPT-4o-mini annotations are added. This clearly shows that the high-quality annotations from GPT-4o are critical to the success of COPELLA.

Qualitative Analysis In Table 7, we compare GPT-4o and GPT-4o-mini event annotations. For this analysis, we used sentences sampled from the MAVEN ARG dataset, as this allows us to compare the annotations to a ground truth. GPT-4o consistently provides more informative descriptions. The trigger descriptions provided by GPT-4o-mini are generic, providing little detail beyond what can be immediately inferred from the trigger phrase. For instance, in the second sentence, GPT-4o correctly frames “bogeyed” as a turning point, describing how “Coody lost strokes on hole 14, reducing his lead.” In contrast, GPT-4o-mini gives a vague description: “a golf event where someone bogeyed".

Sentence	Ground Truth	GPT-4o	GPT-4o-mini
Charles Coody won his only major championship, two strokes ahead of runners-up Johnny Miller and Jack Nicklaus.	**Triggers (span, type):** (won, getting) **Arguments (span, role):** (Charles Coody, agent)	**Triggers (span, description):** (won, Charles Coody achieved victory.) **Arguments (span, role, description):** (Charles Coody, agent, The person who won the event) (Major championship, outcome, The championship title won)	**Triggers (span, description):** (won, A sports competition where someone won.) **Arguments (span, role, description):** (Charles Coody, participant, A person involved in the event.) (Championship, prize, The prize obtained in the competition.)
Miller was six-under for the Sunday round and, playing two groups ahead of the final two-some, his birdie on 14 would open up a two-shot lead when Coody subsequently bogeyed the hole, but could not hold on to win.	**Triggers:** (lead, scoring) (win, getting) **Arguments:** (Miller, agent) (Two-shot lead, score change)	**Triggers:** (bogeyed, Coody lost strokes on hole 14, reducing his lead.) **Arguments:** (Coody, agent, The golfer who bogeyed.) (Hole 14, location, The hole where the event occurred.)	**Triggers:** (bogeyed, A golf event where someone bogeyed.) **Arguments:** (Coody, player, A golfer in the competition.) (Hole 14, location, The place where the event happened.)
It was a bit of redemption for Coody, who bogeyed the final three holes in 1969 to finish two strokes back.	**Triggers:** (redemption, redemption) (bogeyed, playing) **Arguments:** (Coody, agent) (Bogeyed three holes in 1969, past failure)	**Triggers:** (redemption, Coody redeemed himself after past failure.) **Arguments:** (Coody, agent, The person who redeemed himself.) (Final three holes in 1969, past failure, The event from which Coody redeemed himself.)	**Triggers:** (bogeyed, A golf event where someone bogeyed.) **Arguments:** (Coody, player, A golfer who participated.) (Three holes, score change, The number of holes bogeyed.)

Table 7: Comparison of ground truth annotations with annotations from GPT-4o and GPT-4o-mini on sentences sampled from the MAVEN-ARG dataset.

5. Conclusions

LLMs do not consistently outperform BERT-based methods on event extraction tasks. In this paper, we tested the hypothesis that they can nonetheless provide us with a useful distant supervision signal. In particular, we used an LLM to select event triggers and argument spans in sentences, and to generate descriptions for these spans. We then used the resulting dataset for pre-training a BERT-based event encoder. In our experiments, we found this strategy to be highly successful when GPT-4o was used for annotating the pre-training dataset, while being largely unsuccessful with GPT-4o-mini.

As LLMs continue to improve, we can expect the quality of LLM-based event extraction methods to improve as well. However, the use of BERT-based encoders will likely remain appealing, given their significant efficiency advantage. Moreover, better LLMs should also lead to higher-quality annotations, and thus improve the effectiveness of the proposed strategy. Finally, we note that due to cost constraints, we could only test our strategy up to 150k sentences, while further gains are likely to be possible with larger pre-training datasets.

Limitations

A key limitation of our approach is that we obtain LLM annotations at sentence level, whereas some of the most challenging problems in event extraction occur at the document level, e.g. identifying arguments which are mentioned in different sentences and inferring arguments which are only mentioned implicitly. In a preliminary analysis, we found the annotations at sentence level to be of higher quality, which is why we did not pursue document-level LLM annotations for this work. Among others, with the document-level annotations, we found it harder to make the LLM provide the exact spans (rather than paraphrases) and we found the annotations to be less comprehensive, with many events not being annotated. We also noticed a higher rate of hallucinated argument spans, although further work is clearly needed to determine whether these issues could be avoided (e.g. by using a different prompt). In this paper, we did not include a comprehensive comparison of the quality of the annotations obtained from different LLMs, apart from GPT-4o-mini, as our focus was on showing the feasibility of using LLMs for this purpose, and on analyzing the effectiveness of our proposed pre-training strategy.

Acknowledgments This work has been supported by EPSRC grants EP/W003309/1 and EP/Y028805/1.

6. Bibliographical References

Jun Araki and Teruko Mitamura. 2018. Open-domain event detection using distant supervision. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 878–891, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *CoRR*, abs/2404.05961.

Martin Juan José Bucher and Marco Martini. 2024. Fine-tuned'small'llms (still) significantly outperform zero-shot generative ai models in text classification. *arXiv preprint arXiv:2406.08660*.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–419, Vancouver, Canada. Association for Computational Linguistics.

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1267–1276, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

Aleksandra Edwards and Jose Camacho-Collados. 2024. Language models for text classification: Is in-context learning enough? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10058–10072, Torino, Italia. ELRA and ICCL.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jun Gao, Huan Zhao, Yice Zhang, Wei Wang, Changlong Yu, and Ruifeng Xu. 2023. Benchmarking large language models with augmented instructions for fine-grained information extraction. *arXiv preprint arXiv:2310.05092*.

Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023. Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors. *arXiv preprint arXiv:2305.14450*.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. DEGREE: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1890–1908, Seattle, United States. Association for Computational Linguistics.

Kuan-Hao Huang, I-Hung Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, and Heng Ji. 2024. TextEE: Benchmark, reevaluation, reflections, and future challenges in event extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12804–12825, Bangkok,

Thailand. Association for Computational Linguistics.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann.

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. *arXiv preprint arXiv:2304.11633*.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020. Event extraction as multi-turn question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 829–838, Online. Association for Computational Linguistics.

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

Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Lixiang Lixiang, Zhilei Hu, Long Bai, Wei Li, Yidan Liu, Pan Yang, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2024. KnowCoder: Coding structured knowledge into LLMs for universal information extraction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8758–8779, Bangkok, Thailand. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online. Association for Computational Linguistics.

Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022. Dynamic prefix-tuning for generative template-based event extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5216–5228, Dublin, Ireland. Association for Computational Linguistics.

Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2023. Universal information extraction as unified semantic matching. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 13318–13326. AAAI Press.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6759–6774, Dublin, Ireland. Association for Computational Linguistics.

Zihao Meng, Tao Liu, Heng Zhang, Kai Feng, and Peng Zhao. 2024. CEAN: Contrastive event aggregation network with LLM-based augmentation for event extraction. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 321–333, St. Julian's, Malta. Association for Computational Linguistics.

Thien Huu Nguyen and Thien Minh Nguyen. 2018. One for all: Neural joint modeling of entities and events. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 6851–6858.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021*.

Hao Peng, Xiaozhi Wang, Jianhui Chen, Weikai Li, Yunjia Qi, Zimu Wang, Zhili Wu, Kaisheng Zeng, Bin Xu, Lei Hou, and Juanzi Li. 2023. When does in-context learning fall short and why? A study on specification-heavy tasks. *CoRR*, abs/2311.08993.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Kevin Reschke, Martin Jankowiak, Mihai Surdeanu, Christopher Manning, and Daniel Jurafsky. 2014. Event extraction using distant supervision. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4527–4531, Reykjavik, Iceland. European Language Resources Association (ELRA).

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III*, volume 6323 of *Lecture Notes in Computer Science*, pages 148–163. Springer.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. Gollie: Annotation guidelines improve zero-shot information-extraction. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Omar Sharif, Madhusudan Basak, Tanzia Parvin, Ava Scharfstein, Alphonso Bradham, Jacob T Borodovsky, Sarah E Lord, and Sarah M Preum. 2024. Characterizing information seeking events in health-related social discourse. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22350–22358.

Saurabh Srivastava, Sweta Pati, and Ziyu Yao. 2025. Instruction-tuning LLMs for event extraction with annotation guidelines. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13055–13071, Vienna, Austria. Association for Computational Linguistics.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. 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 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022a. DeepStruct: Pretraining of language models for structure prediction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823, Dublin, Ireland. Association for Computational Linguistics.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023. Instructuie: Multi-task instruction tuning for unified information extraction. *CoRR*, abs/2304.08085.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019. Adversarial training for weakly supervised event detection. 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 998–1008, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiaozhi Wang, Hao Peng, Yong Guan, Kaisheng Zeng, Jianhui Chen, Lei Hou, Xu Han, Yankai Lin, Zhiyuan Liu, Ruobing Xie, Jie Zhou, and Juanzi Li. 2024. MAVEN-ARG: Completing the puzzle of all-in-one event understanding dataset with event argument annotation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4072–4091, Bangkok, Thailand. Association for Computational Linguistics.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671, Online. Association for Computational Linguistics.

Xing David Wang, Ulf Leser, and Leon Weber. 2022b. BEEDS: Large-scale biomedical event extraction using distant supervision and question answering. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 298–309, Dublin, Ireland. Association for Computational Linguistics.

Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. 2021. CLEVE: Contrastive Pre-training for Event Extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6283–6297, Online. Association for Computational Linguistics.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, and Jun Zhao. 2018. DCFEE: A document-level Chinese financial event extraction system based on automatically labeled training data. In *Proceedings of ACL 2018, System Demonstrations*, pages 50–55, Melbourne, Australia. Association for Computational Linguistics.

Ying Zeng, Yansong Feng, Rong Ma, Zheng Wang, Rui Yan, Chongde Shi, and Dongyan Zhao. 2018. Scale up event extraction learning via automatic training data generation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 6045–6052. AAAI Press.

Xinliang Frederick Zhang, Carter Blum, Temma Choji, Shalin Shah, and Alakananda Vempala. 2024. ULTRA: Unleash LLMs' potential for event argument extraction through hierarchical modeling and pair-wise self-refinement. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8172–8185, Bangkok, Thailand. Association for Computational Linguistics.

7. Language Resource References

Christopher Walker, Stephanie Strassel, Julie Medero, Kazuaki Maeda. 2006. *ACE 2005 Multilingual Training Corpus*. Linguistic Data Consortium: LDC2006T06, ISLRN 458-031-085-383-4.

David Graff, Junbo Kong, Ke Chen, Kazuaki Maeda. 2005. *English Gigaword Second Edition*. Linguistic Data Consortium: LDC2005T12, ISLRN 274-788-133-216-1.

A. Details of LLM Data Annotation

To support the pre-training of event extraction encoders, we generated a structured dataset using GPT-4o. This dataset was created from sentences that were sampled from the Gigaword corpus, with events extracted and represented in a structured JSON format. The process involved instructing the language model with a carefully designed prompt that included detailed instructions and examples. For each sentence, the model identified:

- **Event Triggers**: Text spans representing the occurrence of events, including their locations.
- **Event Descriptions**: Brief descriptions summarizing the events.
- **Arguments**: Relevant text spans with roles (e.g., agent, object, location) and descriptions of these roles.

- **Entities**: Key entities involved in the events, along with their roles and contextual descriptions.

This approach allowed us to create a high-quality dataset for training, with diverse and accurate annotations.

Prompt We have used the following prompt to obtain event descriptions.

```
{
    "instruction": "Extract events from a given
    passage and represent them in a structured JSON
    format.",
    "input": "<Input Text>",
    "output": {
        "events": [
            {
                "trigger": {
                    "text": "<Trigger Text>",
                    "location": ["<Start Index>",
                    "<End Index>"]
                },
                "description": "<Event Description>",
                "arguments": [
                    {
                        "text": "<Argument Text>",
                        "location": ["<Start Index>",
                        "<End Index>"],
                        "role": "<Role>",
                        "role_description": "<Role
                        Description>"
                    }
                ],
                "entities": [
                    {
                        "text": "<Entity Text>",
                        "location": ["<Start Index>",
                        "<End Index>"],
                        "role": "<Entity Role>",
                        "role_description": "<Entity
                        Role Description>"
                    }
                ]
            }
        ]
    },
    "examples": [
        {
            "instruction": "Extract events from a
            given passage and represent them in a
            structured JSON format.",
            "input": "John gave Mary a book on
            Tuesday.",
            "output": {
                "events": [
                    {
                        "trigger": {
                            "text": "gave",
                            "location": [5, 9]
                        },
                        "description": "An event
                        where John gives a book to
                        Mary.",
                        "arguments": [
                            {
                                "text": "John",
                                "location": [0, 4],
                                "role": "agent",
                                "role_description":
                                "The person performing
                                the action of giving"
                            },
                            {
                                "text": "Mary",
                                "location": [10, 14],
                                "role": "recipient",
                                "role_description":
                                "The person receiving
                                the book"
                            },
                            {
                                "text": "a book",
                                "location": [15, 21],
                                "role": "object",
                                "role_description":
                                "The item being given"
                            },
                            {
                                "text": "on Tuesday",
                                "location": [22, 32],
                                "role_description":
                                "The time when the
                                event occurred"
                            }
                        ],
                        "entities": [
                            {
                                "text": "John",
                                "location": [0, 4],
                                "role": "agent",
                                "role_description":
                                "The person performing
                                the action of giving"
                            },
                            {
                                "text": "Mary",
                                "location": [10, 14],
                                "role": "recipient",
                                "role_description":
                                "The person receiving
                                the book"
                            }
                        ]
                    }
                ]
            }
        }
    ]
}
```

Example outputs We now provide a few examples of the raw outputs we obtained from GPT-4o. First, we consider the following input:

On Saturday, the French asked the U.N. Security Council to authorize the creation of a safe haven in southwestern Rwanda.

We received the following output.

```
{
    "events": [
        {
            "trigger": {
                "text": "asked",
                "location": [18, 23]
            },
            "description": "The French requested
            authorization from the U.N. Security
            Council for a safe haven in Rwanda.",
            "arguments": [
                {
                    "text": "the French",
                    "location": [12, 22],
                    "role": "proposer",
                    "role_description": "The entity
                    making the request"
                },
                {
                    "text": "the U.N. Security
                    Council",
                    "location": [28, 50],
                    "role": "addressee",
                    "role_description": "The entity
                    being asked for authorization"
                },
                {
                    "text": "to authorize the
                    creation of a safe haven",
                    "location": [51, 91],
                    "role": "action",
                    "role_description": "The action
                    requested by the French"
```

```
            },
            {
                "text": "in southwestern Rwanda",
                "location": [95, 118],
                "role": "location",
                "role_description": "The proposed
                location for the safe haven"
            },
            {
                "text": "On Saturday",
                "location": [0, 10],
                "role": "time",
                "role_description": "When the
                request was made"
            }
        ],
        "entities": [
            {
                "text": "the French",
                "location": [12, 22],
                "role": "proposer",
                "role_description": "The entity
                making the request"
            },
            {
                "text": "U.N. Security Council",
                "location": [28, 50],
                "role": "addressee",
                "role_description": "The entity
                responsible for granting
                authorization"
            },
            {
                "text": "Rwanda",
                "location": [113, 119],
                "role": "country",
                "role_description": "The country
                where the safe haven is proposed"
            }
        ]
    }
  ]
}
```


Next, we consider the following input:

The Patriots picked up Jeff Dellenbach from the Dolphins as an unrestricted free agent and drafted Dave Wohlabaugh of Syracuse in the fourth round.

We obtained the following output:


```
{
    "events": [
        {
            "trigger": {
                "text": "picked up",
                "location": [12, 21]
            },
            "description": "The Patriots acquired
            Jeff Dellenbach from the Dolphins.",
            "arguments": [
                {
                    "text": "The Patriots",
                    "location": [0, 12],
                    "role": "agent",
                    "role_description": "The team
                    acquiring the player"
                },
                {
                    "text": "Jeff Dellenbach",
                    "location": [22, 37],
                    "role": "acquired",
                    "role_description": "The player
                    being acquired"
                },
                {
                    "text": "from the Dolphins",
                    "location": [38, 54],
                    "role": "source",
                    "role_description": "The team
                    from which Jeff Dellenbach was
                    acquired"
                },
                {
                    "text": "as an unrestricted free
                    agent",
                    "location": [58, 89],
                    "role": "manner",
                    "role_description": "The manner
                    in which Jeff Dellenbach was
                    acquired"
                }
            ],
            "entities": [
                {
                    "text": "The Patriots",
                    "location": [0, 12],
                    "role": "agent",
                    "role_description": "The
                    acquiring team"
                },
                {
                    "text": "Jeff Dellenbach",
                    "location": [22, 37],
                    "role": "acquired",
                    "role_description": "The player
                    acquired by the team"
                },
                {
                    "text": "the Dolphins",
                    "location": [43, 54],
                    "role": "source",
                    "role_description": "The former
                    team of the acquired player"
                }
            ]
        },
        {
            "trigger": {
                "text": "drafted",
                "location": [94, 101]
            },
            "description": "The Patriots selected
            Dave Wohlabaugh in the NFL draft.",
            "arguments": [
                {
                    "text": "The Patriots",
                    "location": [0, 12],
                    "role": "agent",
                    "role_description": "The team
                    selecting the player"
                },
                {
                    "text": "Dave Wohlabaugh",
                    "location": [102, 117],
                    "role": "selected",
                    "role_description": "The player
                    being drafted"
                },
                {
                    "text": "of Syracuse",
                    "location": [118, 129],
                    "role": "origin",
                    "role_description": "The college
                    from which the player was drafted"
                },
                {
                    "text": "in the fourth round",
                    "location": [130, 149],
                    "role": "time",
                    "role_description": "The round in
                    which the player was drafted"
                }
            ],
            "entities": [
                {
                    "text": "The Patriots",
                    "location": [0, 12],
                    "role": "agent",
                    "role_description": "The team
                    drafting the player"
                },
                {
                    "text": "Dave Wohlabaugh",
                    "location": [102, 117],
                    "role": "selected",
                    "role_description": "The player
                    drafted by the team"
                }
            ]
```

```
        }
    ]
}
```


As a final example, we consider the following input:

In legal papers aimed at blocking Dickstein from recruiting other shareholders to his cause, the company accuses Dickstein of sloppy regulatory filings and dredges up old questions about his integrity stemming from questionable commodity trades in the late 1980s.

We obtained the following output:


```
{
    "events": [
        {
            "trigger": {
                "text": "aimed at blocking",
                "location": [17, 33]
            },
            "description": "The purpose of the legal
            papers is to prevent Dickstein from
            recruiting shareholders.",
            "arguments": [
                {
                    "text": "legal papers",
                    "location": [3, 15],
                    "role": "instrument",
                    "role_description": "The
                    documents being used to block
                    Dickstein"
                },
                {
                    "text": "Dickstein",
                    "location": [34, 43],
                    "role": "target",
                    "role_description": "The person
                    being prevented from recruiting"
                },
                {
                    "text": "recruiting other
                    shareholders to his cause",
                    "location": [49, 89],
                    "role": "action",
                    "role_description": "The activity
                    Dickstein is being blocked from
                    doing"
                }
            ],
            "entities": [
                {
                    "text": "Dickstein",
                    "location": [34, 43],
                    "role": "target",
                    "role_description": "The
                    individual facing legal blockage"
                }
            ]
        },
        {
            "trigger": {
                "text": "accuses",
                "location": [106, 113]
            },
            "description": "The company accuses
            Dickstein of sloppy regulatory filings.",
            "arguments": [
                {
                    "text": "the company",
                    "location": [90, 100],
                    "role": "accuser",
                    "role_description": "The entity
                    making the accusation"
                },
                {
                    "text": "Dickstein",
                    "location": [114, 123],
                    "role": "accused",
                    "role_description": "The person
                    being accused"
                },
                {
                    "text": "sloppy regulatory
                    filings",
                    "location": [127, 151],
                    "role": "accusation",
                    "role_description": "The basis of
                    the company's accusation"
                }
            ],
            "entities": [
                {
                    "text": "Dickstein",
                    "location": [114, 123],
                    "role": "accused",
                    "role_description": "The
                    individual accused by the company"
                }
            ]
        },
        {
            "trigger": {
                "text": "dredges up",
                "location": [156, 166]
            },
            "description": "The company brings up
            past questions about Dickstein's
            integrity.",
            "arguments": [
                {
                    "text": "the company",
                    "location": [90, 100],
                    "role": "agent",
                    "role_description": "The entity
                    bringing up past issues"
                },
                {
                    "text": "old questions about his
                    integrity",
                    "location": [167, 201],
                    "role": "object",
                    "role_description": "The past
                    issues being raised"
                },
                {
                    "text": "stemming from
                    questionable commodity trades in
                    the late 1980s",
                    "location": [202, 261],
                    "role": "reason",
                    "role_description": "The origin
                    of the questions about integrity"
                }
            ],
            "entities": [
                {
                    "text": "Dickstein",
                    "location": [114, 123],
                    "role": "subject",
                    "role_description": "The person
                    whose integrity is questioned"
                }
            ]
        }
    ]
}
```


The model successfully identifies the three events, and accurately describes the different arguments.

B. Qualitative Analysis of GPT-4o Annotations

Table 8 contains examples of annotations that were obtained from GPT-4o. For each input sentence, we show the triggers and arguments that were identified, as well as the corresponding descriptions that were provided by the model. Several observations can be made. First, the model is overall highly successful in identifying the events and arguments, even in complex sentences (such as in the second

Input: *As the French were leaving Butare on Sunday at midday, rebel forces were within 500 yards of the city.*	
Trigger: were leaving	**Descr.:** French troops were departing from Butare.
Arg.: the French	**Descr.:** The group that was leaving
Arg.: Butare	**Descr.:** The place being left
Arg.: on Sunday at midday	**Descr.:** When the departure was happening
Trigger: were within	**Descr.:** Rebel forces were very close to Butare.
Arg.: rebel forces	**Descr.:** The group approaching the city
Arg.: 500 yards of the city	**Descr.:** The distance from Butare
Input: *The Patriots picked up Jeff Dellenbach from the Dolphins as an unrestricted free agent and drafted Dave Wohlabaugh of Syracuse in the fourth round.*	
Trigger: picked up	**Descr.:** The Patriots acquired Jeff Dellenbach from the Dolphins.
Arg.: The Patriots	**Descr.:** The team acquiring the player
Arg.: Jeff Dellenbach	**Descr.:** The player being acquired
Arg.: from the Dolphins	**Descr.:** The team from which Jeff Dellenbach was acquired
Arg.: as an unrestricted free agent	**Descr.:** The manner in which Jeff Dellenbach was acquired
Trigger: drafted	**Descr.:** The Patriots selected Dave Wohlabaugh in the NFL draft.
Arg.: The Patriots	**Descr.:** The team selecting the player
Arg.: Dave Wohlabaugh	**Descr.:** The player being drafted
Arg.: of Syracuse	**Descr.:** The college from which the player was drafted
Arg.: in the fourth round	**Descr.:** The round in which the player was drafted
Input: *Microsoft got into the hardware business in 1982.*	
Trigger: got into	**Descr.:** Microsoft entered the hardware business.
Arg.: Microsoft	**Descr.:** The company entering the hardware business
Arg.: the hardware business	**Descr.:** The business area Microsoft entered
Arg.: in 1982	**Descr.:** The time period when Microsoft entered the hardware business
Input: *Police Commissioner William J. Bratton has said DelDebbio was not motivated by a racial hatred that caused him to fire at a prostrate man, a violation of police rules. He said other witnesses have said the first shots into Officer Robinson might have spun him around so that the subsequent shots entered his back.*	
Trigger: has said	**Descr.:** Police Commissioner William J. Bratton makes a statement about DelDebbio's motivation.
Arg.: Police Commissioner ... Bratton	**Descr.:** The person making the statement
Arg.: DelDebbio was not motivated ... hatred	**Descr.:** The content of the statement about DelDebbio's motivation
Trigger: caused	**Descr.:** DelDebbio fired a weapon at a prostrate man, which violates police rules.
Arg.: him	**Descr.:** The person who fired the weapon, referring to DelDebbio
Arg.: fire at a prostrate man	**Descr.:** What DelDebbio did that was a violation
Arg.: a violation of police rules	**Descr.:** The result of DelDebbio's action
Trigger: said	**Descr.:** Other witnesses provided a statement about the incident involving Officer Robinson.
Arg.: other witnesses	**Descr.:** The people who witnessed and reported on the incident
Arg.: the first shots into Officer ... around	**Descr.:** The content of the statement made by other witnesses
Trigger: entered	**Descr.:** Subsequent shots entered Officer Robinson's back.
Arg.: subsequent shots	**Descr.:** The bullets that hit Officer Robinson
Arg.: his back	**Descr.:** The part of Officer Robinson's body hit by the shots

Table 8: Examples of GPT-4o generated annotations.

and fourth example). Moreover, the descriptions of the arguments are usually informative and accurate. When it comes the triggers, the quality of the annotations is slightly less consistent. For instance, *caused* is incorrectly identified as an event trigger in the fourth example. Moreover, the descriptions that are provided for event triggers are often simple paraphrases of (parts of) the original sentence. In the second example, for the trigger *drafted*, the description clarifies that the event is about the NFL draft, despite this not being explicitly stated in the sentence. When providing argument descriptions, the model also often provides additional context. For instance, in the second example, it is speci-

Input: *The man, Kei Sunada, 22, had been living in New York for only four months when he was shot in what the police say may have been a robbery in his apartment building in the Lefrak City complex. A chef in training at Nakagawa restaurant in midtown Manhattan, Sunada had come to the United States to pursue his interests in race car driving, boxing and cooking.*	
Trigger: had been living	**Descr.:** Kei Sunada was residing in New York.
Arg.: Kei Sunada	**Descr.:** The person living in New York
Arg.: in New York	**Descr.:** Where Kei Sunada was living
Arg.: for only four months	**Descr.:** The length of time Kei Sunada had been living in New York
Trigger: was shot	**Descr.:** Kei Sunada was shot in a possible robbery.
Arg.: Kei Sunada	**Descr.:** The person who was shot
Arg.: in what the police say ... a robbery	**Descr.:** The context in which the shooting occurred
Arg.: in his apartment ... Lefrak City complex	**Descr.:** Where the shooting took place
Trigger: had come	**Descr.:** Kei Sunada arrived in the United States to pursue interests.
Arg.: Kei Sunada	**Descr.:** The person arriving in the United States
Arg.: to the United States	**Descr.:** Kei Sunada's destination
Arg.: to pursue his interests ... and cooking	**Descr.:** The reasons for Kei Sunada's move to the United States
Input: *In legal papers aimed at blocking Dickstein from recruiting other shareholders to his cause, the company accuses Dickstein of sloppy regulatory filings and dredges up old questions about his integrity stemming from questionable commodity trades in the late 1980s.*	
Trigger: aimed at blocking	**Descr.:** The purpose of the legal papers is to prevent Dickstein from recruiting shareholders.
Arg.: legal papers	**Descr.:** The documents being used to block Dickstein
Arg.: Dickstein	**Descr.:** The person being prevented from recruiting
Arg.: recruiting other shareholders to his cause	**Descr.:** The activity Dickstein is being blocked from doing
Trigger: accuses	**Descr.:** The company accuses Dickstein of sloppy regulatory filings.
Arg.: the company	**Descr.:** The entity making the accusation
Arg.: Dickstein	**Descr.:** The person being accused
Arg.: sloppy regulatory filings	**Descr.:** The basis of the company's accusation
Trigger: dredges up	**Descr.:** The company brings up past questions about Dickstein's integrity.
Arg.: the company	**Descr.:** The entity bringing up past issues
Arg.: old questions about his integrity	**Descr.:** The past issues being raised
Arg.: stemming from ... in the late 1980s	**Descr.:** The origin of the questions about integrity
Input: *The more sophisticated RFLP test could break down genetic codes unique to individuals.*	
Trigger: could break down	**Descr.:** The RFLP test has the potential to decompose genetic codes unique to individuals.
Arg.: The more sophisticated RFLP test	**Descr.:** The test that has the capability to analyze genetic codes
Arg.: genetic codes unique to individuals	**Descr.:** The target of the breakdown by the RFLP test
Input: *This dish from the Ivory Coast is traditionally made with chicken and vegetables cooked together for special occasions.*	
Trigger: is traditionally made	**Descr.:** A dish from the Ivory Coast is customarily prepared with chicken and vegetables for special occasions.
Arg.: This dish from the Ivory Coast	**Descr.:** The dish that is being described
Arg.: with chicken and vegetables	**Descr.:** The main ingredients used in the dish
Arg.: cooked together	**Descr.:** The way the ingredients are prepared
Arg.:for special occasions	**Descr.:** The intended occasions for making the dish
Input: *On Saturday, the French asked the U.N. Security Council to authorize the creation of a safe haven in southwestern Rwanda.*	
Trigger: asked	**Descr.:** The French requested authorization from the U.N. Security Council for a safe haven in Rwanda.
Arg.: the French	**Descr.:** The entity making the request
Arg.: the U.N. Security Council	**Descr.:** The entity being asked for authorization
Arg.: to authorize the creation of a safe haven	**Descr.:** The action requested by the French
Arg.: in southwestern Rwanda	**Descr.:** The proposed location for the safe haven
Arg.: On Saturday	**Descr.:** When the request was made

Table 9: Additional examples of GPT-4o generated annotations.

fied that the Dolphins and Patriots are teams, while in the first example, the ambiguous reference to "the city" is resolved by mentioning "Butare" in the description. However, in the second example, Syracuse is incorrectly referred to as a college, rather than a university. The fourth example shows that some inputs consist of more than one sentence, due to mistakes by the sentence splitter. However, the model is able to successfully deal with such inputs.

Table 9 provides additional examples of GPT-4o generated annotations. The main observations are aligned with those from Table 8. The model was overall successful in interpreting the event structures in the inputs, even for complex sentences, as exemplified in the first two examples. However, some of the annotations are sub-optimal. For instance, in the first example, the span "in what the police say may have been a robbery" is arguably too long (where "a robbery" would have been preferable). In the second example, "the company" is missing as an argument of the first event. Moreover, "stemming from questionable commodity trades in the late 1980s" should not be listed as a separate argument and could have been added to the span "old questions about his integrity" instead. The third and fourth example show cases where the sentence is not actually talking about an event. Since we do not provide the model with annotation guidelines, it is not clear what the right response should be in such cases, and the annotations that were provided are meaningful. In the fifth example, "authorize" might also have been considered as a trigger (although this case is again somewhat unclear in the absence of annotation guidelines).

C. Implementation Details

Models We use the following language models in this paper:

- `bert-base-uncased`[9], available under an Apache 2.0 license;
- `bert-large-uncased`[10], available under an Apache 2.0 license;
- `ModernBERT-base`[11], available under an Apache 2.0 license;
- `roberta-large`[12], available under an MIT license;
- A pre-trained SBERT model[13], available under an Apacho 2.0 license;
- A pre-trained LLM2Vec model based on Llama 3[14], available under an MIT license.

Pre-training Details Table 10 presents the hyperparameters used for both pre-training and fine-tuning of our models.

D. Additional Experiments

D.1. Analysis of Model Variants

Table 11 compares the "average token representation" and "individual token contribution" strategies on MAVEN, to complement our analysis on MAVEN-ARG from Table 6. Consistent with our findings from MAVEN-ARG, we can again see that "individual token contribution" achieves the best results.

D.2. Comparison with GPT-4o-mini

The complete set of results for the MAVEN-ARG, MAVEN, and ACE 2005 datasets is presented in Table 13. In Table 12, we provide additional comparisons between GPT-4o and GPT-4o-mini event annotations, for sentences sampled from the MAVEN ARG dataset. In the second sentence, it is notable that GPT-4o-mini chose the word "contest" as the primary trigger, rather than "won", despite the latter arguably being the most salient event in the sentence.

[9] https://huggingface.co/google-bert/bert-base-uncased

[10] https://huggingface.co/google-bert/bert-large-uncased

[11] https://huggingface.co/answerdotai/ModernBERT-base

[12] https://huggingface.co/FacebookAI/roberta-large

[13] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[14] https://huggingface.co/McGill-NLP/LLM2Vec-Meta-Llama-3-8B-Instruct-mntp

Hyperparameter	Value / Range	Description
Model and Training Setup		
Fine-tuning Dataset	MAVEN-ARG, MAVEN, ACE 2005	Event extraction benchmark datasets
Annotation Source	GPT-4o / GPT-4o-mini	Source of training annotations
Pre-training Data Size	25K / 50K / 100K / 150K	Number of annotated samples used for pre-training
Optimization and Learning Rate		
Batch Size	8 / 16	Samples per batch
Gradient Accumulation Steps	4	Effective batch size multiplier
Learning Rate	1e-5	Initial learning rate for fine-tuning
LR Scheduler	Linear Decay with Warm-up	Schedule for learning rate adaptation
Warm-up Steps	10% of total steps	Gradual learning rate increase for stability
Optimizer	AdamW	Weight decay regularization optimizer
Weight Decay	0.01	Regularization factor
Training Epochs	20	Maximum fine-tuning epochs
Gradient Clipping	1.0	Clipping threshold to prevent exploding gradients
Regularization and Model Parameters		
Dropout Rate	0.1	Dropout probability for hidden layers
Label Smoothing	0.1	Confidence calibration factor
Contrastive Loss Temperature (τ)	0.05	Temperature for contrastive loss
Masking Probability (MLM)	15%	Probability of token masking in MLM
Negative Sampling Ratio	5:1	Ratio of negative samples in contrastive learning
Evaluation and Inference		
Evaluation Metric	F1 Score, Precision, Recall, EM	Metrics used for model evaluation
Inference Batch Size	16	Samples per batch during inference
Checkpointing Strategy	Best F1 Score	Model saving criteria

Table 10: Hyperparameters for pre-training and fine-tuning.

			P	R	F1
25k	Average	BB	66.7	72.2	69.3
		BL	66.5	73.1	69.6
		MB	66.8	72.6	69.6
		RL	67.7	73.5	70.5
	Individual	BB	68.4	74.6	71.4
		BL	68.9	74.6	71.6
		MB	68.7	74.3	71.4
		RL	69.5	75.4	72.3
50k	Average	BB	66.9	72.7	69.7
		BL	67.2	73.3	70.1
		MB	67.6	72.8	70.1
		RL	68.3	74.0	71.0
	Individual	BB	68.8	74.8	71.7
		BL	69.5	75.4	72.3
		MB	69.3	74.7	71.9
		RL	70.2	75.6	72.8
100k	Average	BB	70.9	76.7	73.7
		BL	70.7	76.2	73.3
		MB	71.5	76.9	74.1
		RL	72.1	77.6	74.7
	Individual	BB	72.0	77.2	74.5
		BL	71.1	77.0	73.9
		MB	72.0	77.3	74.6
		RL	72.1	77.9	74.9
150k	Average	BB	71.4	77.0	74.1
		BL	71.5	76.8	74.1
		MB	71.6	77.3	74.3
		RL	72.5	78.2	75.2
	Individual	BB	72.0	77.2	74.5
		BL	71.3	77.4	74.2
		MB	72.2	77.8	74.9
		RL	73.0	78.8	75.8

Table 11: Results for MAVEN with BERT-base (BB), BERT-large (BL), mBERT-base (MB) and RoBERTa-large (RL) encoders, for different variants of COPELLA, using the *Combined* pre-training strategy and LLM2Vec embeddings of the descriptions.

Sentence	Ground Truth	GPT-4o	GPT-4o-mini
Coody, co-leader with Nicklaus entering the round, rebounded from his bogey at 14 with two consecutive birdies and parred the final two holes while Miller, 23, bogeyed two of the last three holes.	**Triggers:** (bogeyed, scoring) (parred, movement) (rebounded, getting) **Arguments:** (Miller, agent) (Coody, agent) (Recovered from bogey, outcome)	**Triggers:** (rebounded, Coody recovered from his bogey with consecutive birdies.) **Arguments:** (Coody, agent, The person who rebounded.) (Two consecutive birdies, outcome, The success after recovery.)	**Triggers:** (bogeyed, A golf event where someone bogeyed.) **Arguments:** (Coody, player, The person who played.) (Three holes, score change, The number of holes bogeyed.)
Dave Stockton won the twelfth Par 3 contest on Wednesday with a score of 23.	**Triggers:** (contest, competition) (won, getting) **Arguments:** (Dave Stockton, agent) (Victory in Par 3 contest, outcome)	**Triggers:** (won, Dave Stockton secured victory in the Par 3 contest.) **Arguments:** (Dave Stockton, agent, The winner of the event.) (Par 3 contest, competition, The event where he won.)	**Triggers:** (contest, A competitive event.) **Arguments:** (Dave Stockton, participant, A competitor in the event.)

Table 12: Comparison of ground truth annotations with annotations from GPT-4o and GPT-4o-mini on sentences sampled from the MAVEN-ARG dataset.

			MAVEN-ARG				MAVEN			ACE 2005	
			P	R	F1	EM	P	R	F1	Trigger-C	Arg-C
25k	GPT-4o	BB	35.7	35.2	35.4	33.3	68.4	74.6	71.4	71.0	56.4
		BL	39.6	38.5	39.0	34.4	68.9	74.6	71.6	70.6	51.7
		MB	36.6	36.0	36.3	33.7	68.7	74.3	71.4	71.2	55.5
		RL	39.8	39.2	39.5	34.6	69.5	75.4	72.3	70.3	53.2
	GPT-4o-mini	BB	34.3	33.8	34.0	31.9	65.7	71.6	68.5	68.2	54.1
		BL	38.0	36.9	37.4	33.0	66.2	71.6	68.7	67.8	49.6
		MB	35.2	34.6	34.9	32.3	66.0	71.4	68.5	68.3	53.3
		RL	38.2	37.6	37.9	33.2	66.7	72.3	69.4	67.5	51.0
50k	GPT-4o	BB	37.2	37.0	37.1	34.2	68.8	74.8	71.7	72.1	58.8
		BL	40.4	40.0	40.2	34.9	69.5	75.4	72.3	71.9	59.0
		MB	39.7	38.4	39.0	36.0	69.3	74.7	71.9	72.1	58.6
		RL	40.9	40.8	40.8	35.9	70.2	75.6	72.8	74.3	59.7
	GPT-4o-mini	BB	35.0	34.8	34.9	32.1	64.7	70.8	67.6	68.0	55.1
		BL	37.7	37.3	37.5	32.6	65.3	71.3	67.9	67.8	55.3
		MB	36.9	35.7	36.3	33.7	65.0	70.9	67.5	67.9	55.0
		RL	38.5	38.4	38.5	33.6	66.0	71.8	68.7	70.0	55.9
100k	GPT-4o	BB	40.0	38.4	39.2	36.9	72.0	77.2	74.5	73.2	59.2
		BL	41.3	40.8	41.0	37.3	71.1	77.0	73.9	73.4	59.6
		MB	40.5	39.1	39.8	37.1	72.0	77.3	74.6	73.1	59.3
		RL	42.9	41.8	42.3	38.6	72.1	77.9	74.9	75.4	60.8
	GPT-4o-mini	BB	36.0	34.6	35.3	33.2	64.8	69.5	67.1	65.9	53.3
		BL	37.4	36.8	37.1	33.6	63.9	69.3	66.5	66.0	53.7
		MB	36.8	35.2	36.0	33.4	64.8	69.6	67.0	65.7	53.4
		RL	38.8	37.8	38.3	34.6	65.0	70.2	67.4	67.7	55.1

Table 13: Experimental comparison between the model pre-trained on GPT-4o annotations and the model pre-trained on GPT-4o-mini annotations.