A Hybrid Detection and Generation Framework with Separate Encoders for Event Extraction

Ge Shi[*★]
Beijing University of Technology
shige@bjut.edu.cn

Yunyue Su[*]
Computer Network Information Center, CAS
suyunyue@cnic.cn

Yongliang Ma
Langboat
mayongliang@langboat.com

Ming Zhou
Langboat
zhouming@langboat.com

Abstract

The event extraction task typically consists of event detection and event argument extraction. Most previous work models these two subtasks with shared representation by multiple classification tasks or a unified generative approach. In this paper, we revisit this pattern and propose to use independent encoders to model event detection and event argument extraction, respectively, and use the output of event detection to construct the input of event argument extraction. In addition, we use token-level features to precisely control the fusion between two encoders to achieve joint bridging training rather than directly reusing representations between different tasks. Through a series of careful experiments, we demonstrate the importance of avoiding feature interference of different tasks and the importance of joint bridging training. We achieved competitive results on standard benchmarks (ACE05-E, ACE05-E$^+$, and ERE-EN) and established a solid baseline.

1 Introduction

Event extraction has always been an important and challenging task in Natural Language Processing (NLP) (Sundheim, 1992). It aims to extract event triggers with specific types and event arguments with correct roles from unstructured plain texts into a structured form, which mostly describes "who, when, where, what, why" and "how" of real-world events that happened (Li et al., 2021a). For example, Figure 1 shows a `Meet` event, triggered by "*met*", which describes the `Entity` "*Kelly*" meet with another `Entity` "*officials*" in `Place` "*Seoul*".

Previous studies can be roughly classified into classification-based and generation-based methods depending on the decoder used. Classification-based method usually divides EE into two subtasks: (1) Event Detection (ED), which identifies event triggers and their types. (2) Event Argument Extraction (EAE) extracts the arguments and their corresponding roles for given event triggers and then models them as classification tasks, either learned in a pipeline framework or a joint formulation. Recently, generation-based event extraction methods have emerged as an alternative to traditional classification-based methods due to their better data-efficient and flexibility to include additional guidance. These methods take a sentence with discrete or continuous prompts as input and use BART-style backbone learning to summarize the sentence into a natural sentence based on a manually designed template. The template is composed of natural utterances describing argument role labels, which can provide rich label semantics, leading to great success in generation-based event extraction.

However, most of these methods simultaneously learn shared representations for ED and EAE. As shown in previous works (Nguyen and Grishman, 2015; Lu et al., 2019), ED relies more on lexical (e.g., lemma, synonyms) and shallow syntactic features (e.g., pos tags, dependent and governor words of trigger words). At the same time, the EAE task focuses more on syntactic dependency features. For example, the dependency path between trigger and arguments (Liu et al., 2018). Simply using shared representations dealing with the two distinct tasks would hurt their performance. This phenomenon is also observed in similar tasks, such as entity relation extraction (Zhong and Chen, 2021), where they use two different BERTs for modeling entity extraction and relation extraction, respectively.

To this end, we propose a simple but empirically powerful hybrid framework for event extraction. We model ED and EAE using separate encoders to avoid feature interference between these two tasks. In addition, we conduct extensive experiments to investigate the difference between classification-based and generation-based methods, and we ob-

[*]Equal contribution. Yunyue Su contributes during her internship at Langboat.
[★]Corresponding author.

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 3163–3180
May 2-6, 2023

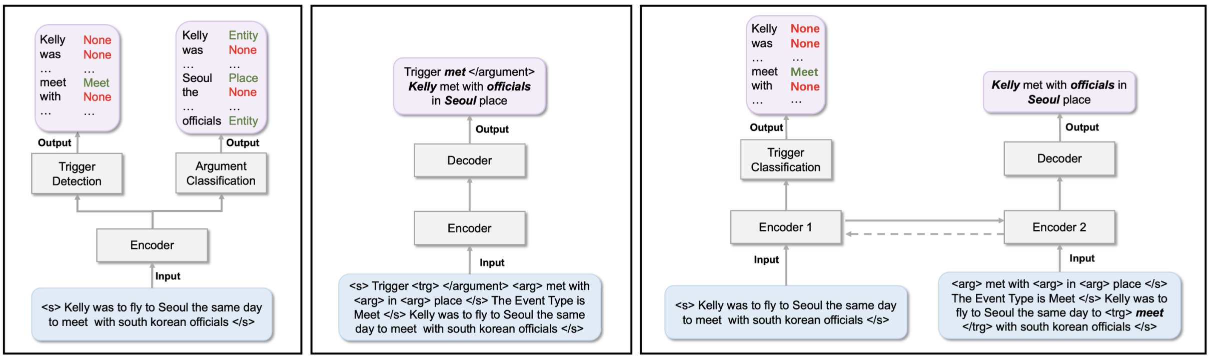

Figure 1: The first two figures are the two major paradigms in the field of event extraction, and the third figure is our paradigm.

serve that (1) Classification-based methods are superior to generation-based methods in modeling token classification tasks. (2) Generation-based methods perform better in modeling EAE because they can capture label semantics. Based on the observations, we instantiate our model with two different decoders: a classification-style decoder for ED and a generation-style decoder for EAE. Finally, to enhance the interaction between these two tasks, we design a bridging mechanism to provide EAE with information derived from ED and a two-stage training method that uses gradients from EAE to guide ED learning. We evaluate our model on three widely used benchmarks, ACE05-E, ACE05-E$^+$, and ERE-EN. Experimental results show that our model establishes the new state-of-the-art on ACE05-E and ACE05-E$^+$, and achieves comparable results on the ERE-EN dataset.

Our contributions can be summarized as follows:

- We first propose a method using separate encoders for modeling event extraction that can avoid feature interference.
- We propose a hybrid classification and generation method that enjoys the advantages of both approaches.
- To model the dependency between ED and EAE, we propose a bridging mechanism and two-stage training method.
- Experimental results show that our proposed method can outperform many strong baselines and achieve new SOTA on ACE05-E, ACE05-E$^+$ [1].

[1]Our codes are publicly available at `https://github.com/OPilgrim/TDE-GTEE`

2 Related Work

Event extraction is usually considered to be composed of two sub-tasks: event detection and event argument extraction. Previous researchers are keen to use a shared encoder to model the contextual representation of different tasks. We group existing event extraction methods into classification-based and generation-based.

Classification-based Method. Classification-based methods tend to model event extraction as a classification task (Mekala and Shang, 2020; Guo et al., 2021; Xiao et al., 2021; Liu et al., 2020; Du and Cardie, 2020b; Li et al., 2020a; Ma et al., 2022) and deal with the recognition of trigger and arguments separately (Ji and Grishman, 2008), e.g., Liang et al. (2020) only consider the event detection and Chen et al. (2015) only consider the extraction of event arguments. Some previous works (Li et al., 2013) have tried to joint training these two tasks to enhance the connection between them, and Yang and Mitchell (2016); Nguyen et al. (2016); Liu et al. (2017, 2018); Lin et al. (2020a) all try to enhance the effect of joint training by adding as much entity and relation information as possible. The difference lies in their shared encoding layers. For example, Liu et al. (2017, 2018) used CNN and Bi-RNN successively, while Wadden et al. (2019a); Lin et al. (2020a) used graph structure. In addition, some works (Ramponi et al., 2020; Du and Cardie, 2020a; Yang et al., 2018) solve the event extraction in sequence labeling manner (Chen et al., 2020; Gui et al., 2020; Jiang et al., 2021) by tagging the sentence only once, which may not solve the overlapping problem.

Generation-based Method. In contrast to classification-based methods, the main goal of generation-based methods is to use a common structure to uniformly model various tasks, including event detection and event argument extraction. The output structure can be a sentence filled with slotted templates (Li et al., 2021b; Du et al., 2022), or some linearly serialized tree structure (Lu et al., 2021a, 2022). Paolini et al. (2021) even constructed an end-to-end translation directly. Generative models can leverage richer prior knowledge. However, the accuracy is not high in classification problems. We think that the generative model may pay more attention to global features and ignores local details. Based on that, several recent works (Hsu et al., 2021; Liu et al., 2022a) use prompt-based approaches to force the model to focus on specific pieces of information to control its output for different event types.

Unlike previous works, we argue that the contextual representation of tasks is different, and sharing one contextual representation will harm the model's performance. So we use independent encoders to learn the contextual representation of each task. In addition, since tasks are not entirely independent, increasing the interaction between tasks will be conducive to improving each other; we achieve it through a bridging mechanism and a two-stage training method.

3 Method

3.1 Problem Definition

The input of the problem is a sentence $\mathcal{C}$ consisting of N tokens $c_1, c_2, ..., c_N$. Let $\mathcal{E} = \{e_1, e_2, ..., e_M\}$ denotes a set of pre-defined event types. The event extraction problem can be decomposed into two sub-tasks:

Event Detection (ED). Event detection aims to identify possible event mentions in the input sequence. We define that each pair of (c_i, e_j) refers to an independent event mention and the event detection is, for each token $c_i \in \mathcal{C}$, to predict an event type $y_e(c_i) \in \mathcal{E}$ or $y_e(c_i) = \epsilon$ representing token c_i is not a trigger. The output of the task is $Y_e = \{(c_i, e_j)|c_i \in \mathcal{C}, e_j \in \mathcal{E}\}$.

Event Argument Extraction (EAE). Event argument extraction aims to identify all entities involved in an event mention (c_i, e_j). Let $\mathbf{s} = \{s_1, s_2, ..., s_t\}$ to be a set of candidate entity spans, and $\mathcal{R}^j$ denotes a set of predefined argument roles in event e_j. The event argument extraction is, for each span $s_i \in \mathbf{s}$, to predict a argument role type $y_r(s_i) \in \mathcal{R}^j$, or span s_i is not an argument belongs to event e_j: $y_r(s_i) = \epsilon$. The output of the task is $Y_r = \{(s_i, r_i, e_j)|s_i \in \mathbf{s}, r_i \in \mathcal{R}^j, e_j \in \mathcal{E}\}$.

3.2 Our Approach

In this section, we introduce our proposed method, HDGSE[3] (the **H**ybrid **D**etection and **G**eneration framework with **S**eparate **E**ncoders for **E**vent Extraction), based on the overall architecture of Figure 2.

Event Trigger Detection. We use BERT as the backbone of our detection model and treat it as a token-level multi-classification task, which makes the model learn the different probabilities of each event type (Li et al., 2021b). As we mentioned in Section 2, the sequence annotation method based on CRF cannot solve the span coverage problem, so we did not implement this scheme. Given the input sequence $\mathcal{C} = \{c_1, ..., c_i, ..., c_N\}$, the detection model will detect all possible trigger tokens $\{c_i, | i \in \{i\}_{i=1}^{N}\}$ and their corresponding event type $\{e_j, | j \in \{j\}_{j=1}^{M}\}$ as mentioned in Formula 1,

$$label_{c_i} = \begin{cases} 0, & c_i \text{ is not a trigger,} \\ j, & c_i \text{ is trigger for event type } j. \end{cases} \quad (1)$$

where N is the length of $\mathcal{C}$ and M is the number of event types in ontology $\mathcal{O}$ [2]. Each pair of $(c_i,\ e_j)$ indicates the hit of an event $\{E_k | k \in \{k\}_{k=1}^{K}\}$, where K denotes the number of events in $\mathcal{C}$. Then the generative model extracts arguments for each event E_k in turn.

Generative Argument Extraction. After detecting the candidate event triggers, the argument extraction task is divided into several subtasks according to the detected triggers and event types, and each subtask is an event mention. We process each event mention independently with a generative approach and insert markers at the input sequence to highlight the trigger. The generative model is based on BART, and the lower part of Figure 2 shows the detailed structure. Specifically, for subtask $\mathcal{S}_{E_k,\mathcal{C}}$, the input $\mathcal{X}$ of the generative model includes the event type aware prompt $\mathcal{P}_{e_j}$ and context $\mathcal{C}' = \{c_1, ..., \text{<trg>}, c_i, \text{</trg>}, ..., c_N\}$, where the trigger c_i is marked by two special tokens "<trg>" and "</trg>" to provide trigger position information for the corresponding subtask. Given

[2]We follow Li et al. (2021c) and reuse RAMS AIDA ontology and the KAIROS ontology as the ontology for ACE05-E, ACE05-E$^+$ and ERE.

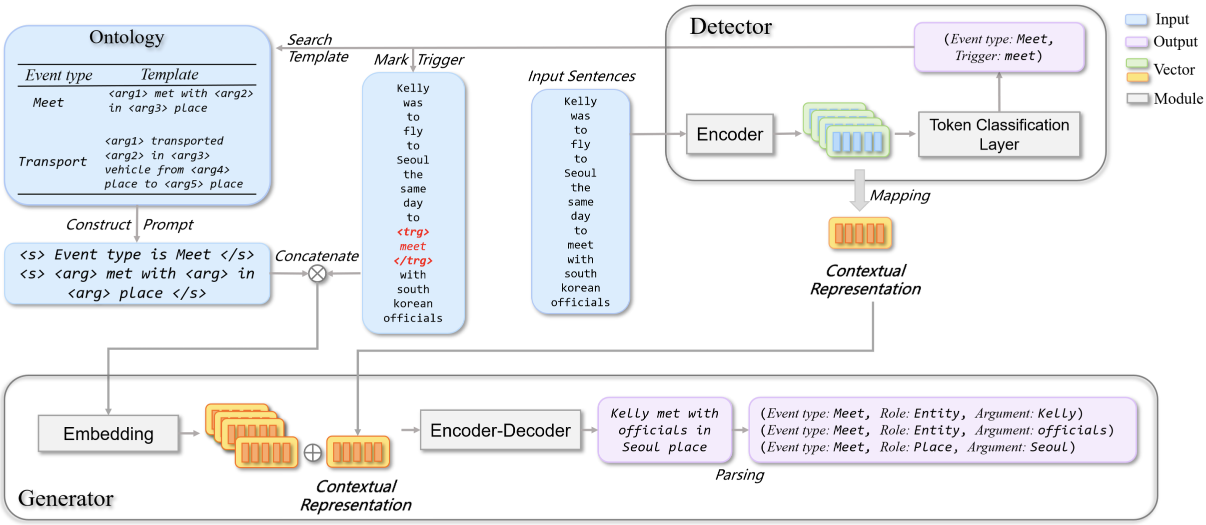

Figure 2: Illustration of our end-to-end joint event extraction framework. Given an input sequence, the detector first detects several candidate triggers and classifies their event types. Then, the generator generates filled template from the trigger-marked input and finally parses the arguments through a deterministic algorithm. During generation, the contextual representation of the trigger that the detector learns is fused into the generator.

the previous generated tokens $y_{<i}$ and the input $\mathcal{X}$, the BART models the conditional probability of selecting the next token y_i as $p(y_i|y_{<i}, \mathcal{X})$, and the entire probability $p(\mathcal{Y}|\mathcal{X})$ is calculated as

$$
\begin{aligned}
p(\mathcal{Y}|\mathcal{X}) &= \prod_{i=1}^{|\mathcal{Y}|} p(y_i|y_{<i}, \mathcal{X}) \\
\mathcal{X} &= [\mathcal{P}_{e_j}; [SEP]; \mathcal{C}'] \\
\mathcal{Y} &= \mathcal{A}_{e_j}
\end{aligned}
\tag{2}
$$

, where $[;]$ denotes the sequence concatenation operation and $[SEP]$ is the corresponding separate marker in the BART, $\mathcal{A}_{e_j}$ is the answered prompt.

Similar to generative template-based method GTEE(Liu et al., 2022b), the prompt $\mathcal{P}_{e_j}$ for subtask $\mathcal{S}_{E_k,\mathcal{C}}$ contains the type instruction $\mathcal{I}_{e_j}$ and the template $\mathcal{T}_{e_j}$. The type instruction is an indication of the event type described by natural language, and the template describes the expected output format, including several placeholders, reflecting how the arguments participant in the event. Take Figure 2 as example, the generative model's input is type instruction "Event type is Meet", template "<arg> met with <arg> in <arg> place", and content concatenated with separator. The ground truth $\mathcal{G}_{e_j}$ is "Kelly met with officials in Seoul place", where the placeholder "<arg>" is replaced by the corresponding argument "Kelly", "officials", and "Seoul". Each event type has its own template and we follow Li et al. (2021b) to reuse the pre-defined argument templates.

Bridging Event Detection and Event Argument Extraction. Our proposal to independently learn contextual representations for ED and EAE does not mean that the two tasks are not connected; argument extraction directly depends on determining event types and triggers. So to enhance the interaction between them, we bridge the two tasks by trigger: first, as mentioned in the previous section, we highlight the trigger in the input of EAE, which provides the location information; the second is to fuse the context information of triggers into the EAE model, which is the focus of this section. Both kinds of information provide the EAE model with prior knowledge of events. Specifically, for the trigger token c_i, its hidden state in BERT's last hidden layer is $\boldsymbol{h}_{c_i}$, and its input embedding in BART is $\boldsymbol{Emb}(c_i)$. A semantic transformation is performed by multiplying the $\boldsymbol{h}_{c_i}$ by the projection matrix $\boldsymbol{P}$ to obtain the projected vector $\boldsymbol{v}_{c_i} = \boldsymbol{h}_{c_i}\boldsymbol{P}$ as contextual representations, where $\boldsymbol{P}$ can be learned by fully connected networks. We refer to this operation as "Mapping", as illustrated in Figure 2. Then, we add $\boldsymbol{v}_{c_i}$ and $\boldsymbol{Emb}(c_i)$ directly. Another appropriate method is sufficient to directly use the vector $\boldsymbol{v}_{c_i}$ to initialize the embedding representation of trigger markers. We conduct comparative experiments against these two appropriates in Section 5.2.

Training and Inference. In this paper, we design a two-stage training approach: (1) In the first step, we first train ED and EAE separately so that they can learn the contextual representation independently. (2) In the second step, to overcome the error propagation problem of the pipeline, we continue to use joint training to optimize the global loss based on the model trained in the first step and use the gradient of EAE to guide the optimization of ED.

Mathematically, The trainable parameters of our model include ϕ and φ, which come from the BERT and BART respectively. The training objective of the detection model is to minimize the focal loss (Lin et al., 2017) between each token's predicted label and the golden label:

$$\mathcal{L}_{\phi}(\mathcal{C}) = \sum_{i=1}^{N} -(1 - p_{c_i,e_j})^{\gamma} log(p_{c_i,e_j}) \quad (3)$$

And the training objective of the argument extraction model is to minimize the negative loglikelihood over all subtasks $\mathcal{S}_{E_k,\mathcal{C}}$ of the input sample $\mathcal{C}$:

$$\begin{aligned} &\mathcal{L}_{\varphi}(\mathcal{C}) = -\sum_{k=1}^{K} log\ p(\mathcal{G}_{e_j,\mathcal{C}'_k} | \mathcal{X}_{e_j,\mathcal{C}'_k}) \\ &\mathcal{X}_{e_j,\mathcal{C}'_k} = [\mathcal{P}_{e_j}; [SEP]; \mathcal{C}'_k] \\ &\mathcal{C}'_k = \{c_1, ..., \text{<trg>}, c_i, \text{</trg>}, ..., c_N\} \end{aligned} \quad (4)$$

Finally, during the joint bridging training phase, the loss of the whole model is:

$$\mathcal{L}(\mathcal{D}) = \sum_{t=1}^{|\mathcal{D}|} (\mathcal{L}_{\phi}(\mathcal{C}_t) + \mathcal{L}_{\varphi}(\mathcal{C}_t)) \quad (5)$$

Implementation details are shown in Appendix B.

4 Experiment

4.1 Setup

Datasets. We evaluate our methods on three widely used event extraction benchmarks, ACE05-E, ACE05-E$^+$ and ERE-EN. Both of the ACE datasets are from the Automatic Content Extraction 2005 (ACE 2005) dataset constructed by Doddington et al. (2004), and the ERE-EN is from ERE dataset (Song et al., 2015). All their details can be found in Appendix A.

Evaluation Metrics. We consider the same criteria following prior works (Liu et al., 2022b; Hsu et al., 2021; Lu et al., 2021b) and report the Precision *P*, Recall *R* and F1 score *F1* of trigger and argument. Meanwhile, we consider that a trigger is correctly identified if its offset matches the ground truth (**Trg-I**) and is correctly classified if its event matches the ground truth as well (**Trg-C**). In the same way, we consider an argument is correctly identified if its offset matches the ground truth (**Arg-I**) and is correctly classified if its event type and role label all matches the ground truth as well (**Arg-C**).

Compared Baselines. We consider several representative works as our baselines, including both classification-based and generation-based methods, and some of their implementation details are listed in Appendix B.

we consider the following classification-based models:

- **DYGIE++**(Wadden et al., 2019b), a BERT-based model learns shared span representations between multi-tasks and updates span representations through dynamic graph propagation layers.
- **GAIL**(Zhang et al., 2019), a RL model jointly extracting entity and event.
- **OneIE**(Lin et al., 2020a), an end-to-end IE system that employs designed global feature and beam search, was state-of-the-art.
- **BERT_QA**(Du and Cardie, 2020c), an MRC-based model views EE tasks as a question-answering problem with multi-turns of separated QA pairs and learns a classifier to indicate the position of the predicted span.
- **MQAEE**:(Li et al., 2020b), a multi-turn question answering system.

We also consider the following generation-based models:

- **TANL**:(Paolini et al., 2021), a method treats EE tasks as translation tasks in a trigger-argument pipeline.
- **BART-GEN**(Paolini et al., 2021), a template-based conditional generation method to generate corresponding arguments in a predefined format.
- **TEXT2EVENT**(Lu et al., 2021b), a sequence-to-structure generation method that converts the input sequence to a tree-like event structure.

- **DEGREE-E2E**(Hsu et al., 2021), an end-to-end conditional generation method that uses natural sentences as discrete prompts, which makes it easier for them to leverage label semantics.

- **GTEE-DYNPREF**(Liu et al., 2022b), an end-to-end conditional genration method with dynamic prompts and trained with prefix-tuning.

4.2 Main Results

For each dataset, we train our model with 5 different random seeds, and report the means of the corresponding results.

Table 1 compares our approach HDGSE[3] with all the baselines on ACE05-E, while Table 2 illustrates the results compared with the state-of-the-art on ACE05-E$^+$ and ERE-EN. As shown, our model achieves strong performance and outperforms all the baselines on two datasets of ACE 2005. At the same time, our model also performs competitively on ERE-EN, second only to GTEE-DYNPREF (Lin et al., 2020b).

For event detection, our model achieves an absolute Trg-C F1 improvement of $+5.8\%$, $+2.9\%$ on ACE05-E and ACE05-E$^+$ respectively compared to DEGREE-E2E (Lin et al., 2020b) and GTEE-DYNPRE (Liu et al., 2022b) that also utilizes joint training but use the generative method for ED, indicating that the classification-based method has more advantages in event detection than the generative-based method. On the other hand, our model also shows a significant improvement over the classification-based methods, e.g., a gain of 4.3% on ACE05-E compared to ONEIE (Lin et al., 2020b). As we will show later in our experiments, part of this improvement is due to the bridging mechanism.

For event argument extraction, our approach outperforms the previous best methods, ONEIE (Lin et al., 2020b) and DEGREE-E2E (Lin et al., 2020b), with absolute Arg-C F1 gains of $+1\%$, $+1.4\%$ on ACE05-E, ACE05-E$^+$ respectively. Moreover, it outperforms the best generative method DEGREE-E2E (Lin et al., 2020b) on ACE05-E and ERE-EN with absolute Arg-C F1 gains of $+2\%$, $+3.9\%$. Such improvements demonstrate the effectiveness of maintaining different contextual representations for ED and EAE, and incorporating trigger information into the EAE model. Although our model performs second only to GTEE-DYNPRE (Liu et al., 2022b) on ERE-EN, it outperforms GTEE-DYNPRE on all other datasets, indicating that our model has better robustness on different datasets.

5 Analysis

5.1 Selection of Task Models

We further investigated the impact of using classification-based or generation-based models for the ED task and the EAE task, respectively, to gain insight into the advantages and disadvantages of these two approaches for event extraction tasks.

Event Detection: Classification vs Generative. We first compare the two paradigms on the event detection task and list the results in Table 3. The experimental details can be found in Appendix B. We observe that the generation-based model is significantly worse than the classification-based model on this task. One possible reason is that generation-based models pay more attention to the global features of sentences and have fewer advantages in ED, which require trigger tokens and their local context. Moreover, the classification-based model can directly provide the location of trigger spans, which is more helpful for EAE. Therefore, the classification-based paradigm is more suitable for the ED task than the generation-based one.

Table 3: The event detection results of the classification-based and generation-based approach.

ED Paradigm	ACE05-E		ACE05-E$^+$		ERE-EN	
	Trg-I	Trg-C	Trg-I	Trg-C	Trg-I	Trg-C
Classification-based	81.9	77.8	80.8	76.9	75.9	65.7
Generation-based	67.8	45.3	66.8	45.6	61.4	37.1

Event Argument Extraction: Classification vs Generative. We eliminated the trigger detection task in order to investigate the impact of different paradigms on event argument extraction. Given golden triggers as input, we implemented several classification and generation paradigm baselines for EAE. The experimental results are shown in Table 4. The generative approach performs as well as the classification-based model under the standard setting (with trigger marker). And several template-based generative approaches, such as GTEE-BASE (Liu et al., 2022b), BART-GEN (Li et al., 2021b), DEGREE-EAE (Lin et al., 2020b) and HDGSE[3], perform significantly better.

In particular, DEGREE-EAE performs best under the gold trigger marker setting. This is mainly due to the fact that DEGREE-EAE incorporates more knowledge of events in the prompt design,

Table 1: Results on ACE05-E for event extraction. The first group is the classification-based methods, and the second is the generation-based methods. For each group, we bold the highest F1 scores for Trg-C and Arg-C, and the second highest is bold in italics.

Model	Trg-C			Arg-C		
	P	R	F1	P	R	F1
classification-based						
DYGIE++ (Wadden et al., 2019b)	-	-	69.7	-	-	48.8
GAIL (Zhang et al., 2019)	74.8	69.4	72.0	61.6	45.7	52.4
ONEIE (Lin et al., 2020b)	-	-	***74.7***	-	-	***56.8***
BERT_QA (Du and Cardie, 2020c)	71.1	73.7	72.3	56.8	50.2	53.3
MQAEE (Li et al., 2020b)	-	-	71.7	-	-	53.4
generation-based						
TANL (Paolini et al., 2021)	-	-	68.5	-	-	48.5
BART-GEN (Li et al., 2021b)	69.5	72.8	71.1	56.0	51.6	53.7
TEXT2EVENT (Lu et al., 2021b)	67.5	71.2	69.2	46.7	53.4	49.8
DEGREE-E2E (Hsu et al., 2021)	-	-	***73.3***	-	-	***55.8***
GTEE-DYNPREF (Liu et al., 2022b)	63.7	84.4	72.6	49.0	64.8	***55.8***
HDGSE[3]	76.1	82.1	**79.0**	55.3	60.4	**57.8**

Table 2: Results on ACE05-E$^+$ and ERE-EN for event extraction. We bold the highest F1 scores for Trg-C and Arg-C, and the second highest is bold in italics.

Model	ACE05-E$^+$						ERE-EN					
	Trg-C			Arg-C			Trg-C			Arg-C		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
ONEIE	72.1	73.6	72.8	55.4	54.3	54.8	58.4	59.9	59.1	51.8	49.2	50.5
TEXT2EVENT	71.2	72.5	71.8	54.0	54.8	54.4	59.2	59.6	59.4	49.4	47.2	48.3
DEGREE-E2E	-	-	70.9	-	-	***56.3***	-	-	57.1	-	-	49.6
GTEE-DYNPREF	67.3	83.0	***74.3***	49.8	60.7	54.7	61.9	72.8	**66.9**	51.9	58.8	**55.1**
HDGSE[3]	75.5	79.0	**77.2**	57.6	57.8	**57.7**	64.5	67.9	***66.1***	54.5	52.6	***53.5***

such as "Event Type Description" and "Event Keywords". Interestingly, our final results on EE are better than DEGREE-EAE because (1) DEGREE-EAE uses a generative paradigm in the event detection task, (2) shares the contextual representation of the encoder between two tasks, which indirectly proves the correctness of our hypothesis.

5.2 Effect of Bridging Mechanisms

We mentioned in Section 3.2 that trigger marker and contextual representation fusion were used to establish a bridge connection for the two independent encoders. This section will look closely at these two components to see how they affect our model.

We remove the possible connection modules between two independent encoders under the settings of Joint and Pipeline, respectively, and present the experimental results in Table 5. It can be seen that removing the trigger marker causes significant damage to the model under both training paradigms. Although contextual representation can also improve the model's performance, the overall improvement space is not as ample as the trigger marker. Further, when we remove both of them, as shown in Table 5, the F1 score of Trg-C remains at a very high level for the three datasets, which are still SOTA at ACE05-E and ACE05-E$^+$. However, at the same time, the F1 of Arg-C is significantly reduced and no longer SOTA. These phenomena show that the effect of ED representing the upper bound of EAE and the bridging mechanism can help EAE approach this upper bound and even improve the result of ED in reverse. That is where the main contribution of the bridging mechanism lies.

From another point of view, when only comparing the training paradigms, it can be found that loss sharing during joint training can significantly improve the model's overall performance, so the Joint results are generally better than those of Pipeline, which proves the effectiveness of our two-stage training program.

We also design several contrast schemes for the fusion way of contextual representation. In Section 3.2, we discussed two approaches, one is to assign the mapped contextual representation to the trigger

Table 4: Results of event argument extraction. Models predict arguments based on the given gold triggers. *We report the numbers from the original paper. †We reproduce the results.

Model	Type	ACE05-E		ACE05-E$^+$		ERE-EN	
		Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
DyGIE++*	Cls	66.2	60.7	-	-	-	-
BERT_QA*	Cls	68.2	65.4	-	-	-	-
OneIE*	Cls	73.2	69.3	***73.3***	***70.6***	75.3	70.0
GTEE-BASE †	Gen	70.1	67.2	67.3	63.6	72.3	66.8
BART-GEN †	Gen	66.9	66.7	70.0	66.8	74.6	69.2
TANL*	Gen	65.9	61.0	66.3	62.3	75.6	69.6
DEGREE-EAE*	Gen	**76.0**	**73.5**	**75.2**	**73.0**	**80.2**	**76.3**
HDGSE^3	Gen	***73.8***	***70.2***	72.1	69.0	***76.4***	***72.0***

Table 5: Ablation study for the effectiveness of trigger marker and fused contextual representation.

Insert Setting (Trained ED)	ACE05-E				ACE05-E$^+$				ERE-EN			
	Trg-I	Trg-C	Arg-I	Arg-C	Trg-I	Trg-C	Arg-I	Arg-C	Trg-I	Trg-C	Arg-I	Arg-C
HDGSE^3 (Joint)	83.0	79.0	60.1	57.8	81.1	77.2	60.2	58.2	76.4	66.1	56.5	53.5
- remove marker trigger	83.0	78.2	58.8	56.9	80.8	76.9	57.6	55.7	75.1	64.9	53.8	50.2
- remove context fusion	82.1	78.0	59.1	56.6	80.8	76.9	58.2	56.1	75.2	65.3	55.8	52.4
- remove both	82.1	78.0	58.6	56.1	80.8	76.9	57.6	55.6	75.0	64.7	53.5	49.9
HDGSE^3 (Pipeline)	77.9	74.2	55.4	53.5	79.2	75.7	56.0	53.3	73.4	62.1	52.0	48.8
- remove marker trigger	77.9	74.2	40.5	38.6	79.2	75.7	46.5	44.7	73.4	62.1	29.7	28.6

marker, and the other is to directly add with the trigger representation learned by the language model during pre-training. In addition to the above two, we further explore what results can be obtained by directly fusing the contextual representation without mapping. Table 6 lists the experimental results. Note that directly fusing the contextual representation of ED and EAE without mapping causes significant damage to the model, which is even worse than the direct deletion of the contextual representation in Table 5. That proves the contextual representations learned by ED and EAE are different and preferably not directly shared. On the other hand, the performance of mapped contextual representations is almost the same regardless of whether they are fused with trigger markers or triggers. We believe this is because contextual representations provide more semantic information, which is not affected by the difference in fusion objects.

Table 6: Study on the fusion form of contextual representation. Models predict arguments based on the predicted trigger.

HDGSE^3 (Joint)	ACE05-E		ACE05-E$^+$		ERE-EN	
	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
trigger marker	60.3	57.8	59.5	57.3	56.5	53.5
- w/o mapping	59.0	56.5	59.9	57.7	55.1	51.7
trigger	60.1	57.8	60.2	58.2	56.5	53.5
- w/o mapping	58.6	55.4	58.0	55.5	55.0	50.2

5.3 Prompts and Templates

Generative-based event extraction methods tend to be sensitive to the prompts and templates used (Liu et al., 2022a). Since our model adopts a generative method for EAE, we further investigated the robustness of our model when using different prompts and templates.

Necessity of Type Instruction. We first consider replacing the static type instruction such as "*The Event Type is Meet*" but still providing explicit event type information to the model. So we refer to Zhong and Chen (2021) and use <trigger:Event type> and </trigger:Event type> instead of the original type instruction. The experimental results in Table 7 show that using sentences in natural language to describe event types will perform better than replacing them with tokens. Therefore, we still keep this setting in our experiments.

Table 7: Study on the necessity of type instruction. r/ stands for replace, and we replaced type instruction with <trigger:Event type> and </trigger:Event type>.

HDGSE^3-EAE (Gold Trg)	ACE05-E		ACE05-E$^+$		ERE-EN	
	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
w/ Type Instruction	73.8	70.2	72.1	69.0	76.4	72.0
r/ Type Instruction	72.8	69.9	71.6	67.4	75.0	69.4

Sensitivity to Template Design. Our method requires templates with slotted values to assist

EAE, so we designed several templates to explore whether the model is robust under different template Settings. We designed three types of templates from low to high semantic integrity and the detailed construction details can be found in Appendix C. We put all three types of templates involved in ACE 2005 and ERE-EN in Tables 13 and 14, and the Table 8 show the experimental results. Templates without semantics perform worst, indicating that the model is still sensitive to the template's design. However, the weak and strong semantic integrity results are close, indicating that the model still has good robustness to sentences with certain linguistic logic. Weak semantic integrity templates can ensure the model's performance, whether manually designed or model-generated. The experiments in this paper are all done based on templates with weak semantic integrity, and we leave generating templates from models for the future.

Table 8: Study on the effect of different template constructing rules. Models predict arguments based on the given gold trigger.

HDGSE^3-EAE (Gold Trg)	ACE05-E		ACE05-E$^+$		ERE-EN	
	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
No semantic integrity	67.7	64.2	66.3	63.0	74.9	70.2
Weak semantic integrity	73.8	70.2	72.1	69.0	76.4	72.0
Strong semantic integrity	73.1	70.4	70.7	69.1	76.6	71.5

6 Conclusion

In this paper, we revisit the classification-based and generation-based event extraction methods and empirically propose a simple but robust hybrid event extraction scheme. Our model learns two independent encoders for event detection and event argument extraction and uses simple trigger marker and contextual representation fusion to bridge training jointly, for which we devise a two-stage training approach. We conduct extensive analyses to understand the superior performance of our approach. These analyses verify the effectiveness of using the classification model and the generative model to learn the contextual representation of event detection and event argument extraction separately and validate the importance of taking the result of event detection as the input of event argument extraction. We hope this simple model will serve as a strong benchmark for end-to-end event extraction and make us rethink the value of a shared representation of multi-tasks.

Limitations

Our findings in this paper only verified in event extraction. It will be more exciting and valuable if migrated to other multi-task problems. We will leave that for future work.

Acknowledgements

We thank the anonymous reviewers for their valuable comments and suggestions. This work is supported by National Natural Science Foundation of China (No. 62106010).

References

Luoxin Chen, Weitong Ruan, Xinyue Liu, and Jianhua Lu. 2020. SeqVAT: Virtual adversarial training for semi-supervised sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8801–8811, Online. Association for Computational Linguistics.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China. 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. 2020a. Document-level event role filler extraction using multi-granularity contextualized encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8010–8020, Online. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020b. 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.

Xinya Du and Claire Cardie. 2020c. Event extraction by answering (almost) natural questions. *arXiv preprint arXiv:2004.13625*.

Xinya Du, Sha Li, and Heng Ji. 2022. Dynamic global memory for document-level argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers), pages 5264–5275, Dublin, Ireland. Association for Computational Linguistics.

Tao Gui, Jiacheng Ye, Qi Zhang, Zhengyan Li, Zichu Fei, Yeyun Gong, and Xuanjing Huang. 2020. Uncertainty-aware label refinement for sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2316–2326, Online. Association for Computational Linguistics.

Biyang Guo, Songqiao Han, Xiao Han, Hailiang Huang, and Ting Lu. 2021. Label confusion learning to enhance text classification models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12929–12936.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

I Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, et al. 2021. Degree: A data-efficient generative event extraction model. *arXiv preprint arXiv:2108.12724*.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, Ohio. Association for Computational Linguistics.

Ting Jiang, Deqing Wang, Leilei Sun, Huayi Yang, Zhengyang Zhao, and Fuzhen Zhuang. 2021. Lightxml: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7987–7994.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020a. 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.

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

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics.

Qian Li, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng, Shu Guo, Lihong Wang, Amin Beheshti, et al. 2021a. A compact survey on event extraction: Approaches and applications. *arXiv preprint arXiv:2107.02126*.

Sha Li, Heng Ji, and Jiawei Han. 2021b. Document-level event argument extraction by conditional generation. *arXiv preprint arXiv:2104.05919*.

Sha Li, Heng Ji, and Jiawei Han. 2021c. 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.

Xin Liang, Dawei Cheng, Fangzhou Yang, Yifeng Luo, Weining Qian, and Aoying Zhou. 2020. F-hmtc: Detecting financial events for investment decisions based on neural hierarchical multi-label text classification. In *IJCAI*, pages 4490–4496.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020a. 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.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020b. 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.

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.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Vancouver, Canada. Association for Computational Linguistics.

Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022a. 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.

Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022b. Dynamic prefix-tuning for generative template-based event extraction. *arXiv preprint arXiv:2205.06166*.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the*

2018 Conference on Empirical Methods in Natural Language Processing, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization (2017). *arXiv preprint arXiv:1711.05101*.

Yaojie Lu, Hongyu Lin, Xianpei Han, and Le Sun. 2019. Distilling discrimination and generalization knowledge for event detection via delta-representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4366–4376, Florence, Italy. Association for Computational Linguistics.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021a. 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, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021b. Text2event: Controllable sequence-to-structure generation for end-to-end event extraction. *arXiv preprint arXiv:2106.09232*.

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. *arXiv preprint arXiv:2203.12277*.

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.

Dheeraj Mekala and Jingbo Shang. 2020. Contextualized weak supervision for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 323–333, Online. Association for Computational Linguistics.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 365–371, Beijing, China. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2101.05779*.

Alan Ramponi, Rob van der Goot, Rosario Lombardo, and Barbara Plank. 2020. Biomedical event extraction as sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5357–5367, Online. Association for Computational Linguistics.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ERE: Annotation of entities, relations, and events. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98, Denver, Colorado. Association for Computational Linguistics.

Beth M. Sundheim. 1992. Overview of the fourth Message Understanding Evaluation and Conference. In *Fourth Message Uunderstanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992.*

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019a. 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.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019b. Entity, relation, and event extraction with contextualized span representations. *arXiv preprint arXiv:1909.03546*.

Sijia Wang, Mo Yu, and Lifu Huang. 2022. The art of prompting: Event detection based on type specific prompts. *arXiv preprint arXiv:2204.07241*.

Lin Xiao, Xiangliang Zhang, Liping Jing, Chi Huang, and Mingyang Song. 2021. Does head label help for long-tailed multi-label text classification. *arXiv preprint arXiv:2101.09704*.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299, San Diego, California. Association for Computational Linguistics.

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.

Tongtao Zhang, Heng Ji, and Avirup Sil. 2019. Joint entity and event extraction with generative adversarial imitation learning. *Data Intelligence*, 1(2):99–120.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

A Datasets

ACE 2005 has 599 annotated English documents, 33 event types, and 22 argument roles. ERE contains 458 English documents, 38 event types, and 21 argument roles. Following previous work (Zhang et al., 2019; Wadden et al., 2019b; Du and Cardie, 2020c; Lin et al., 2020b; Lu et al., 2021b; Hsu et al., 2021), we use the same preprocess method and obtain three datasets, ACE05-E, ACE05-E$^+$, and ERE-EN, shown in Table 9. Compared to ACE05-E, both ACE05-E$^+$ and ERE-EN contain pronoun roles and multi-token event triggers.

Table 9: Dataset statistics.

Dataset	Split	Sents	Events	Roles
ACE05-E	Train	17,172	4202	4859
	Dev	923	450	605
	Test	832	403	576
ACE05-E$^+$	Train	19,216	4419	6607
	Dev	901	468	759
	Test	676	424	689
ERE-EN	Train	14,736	6208	8924
	Dev	1209	525	730
	Test	1163	551	822

B More Implementation details

Main Settings.

Table 10: Hyperparameters for two-stage of training, with the first phase being separate independent training for the two tasks and the second phase being joint bridging training.

Name	Independent Training		Joint Bridge Training
	ED (BERT)	EAE (BART)	EE (BERT+BART)
Learning rate	2e-5	2e-5	5e-7
Batch size	1*32	1*32	1*32
Epochs	40	40	30
Max sequence length	185 | 325	Id.	Id.
Max output length	-	50	Id.
Weight decay	1e-5	1e-5	1e-5
Gradient clip	5.0	5.0	5.0
Warm-up ratio	10%	10%	10%
Loss	Focal	Cross-entropy	-
Focal gamma	3	-	-
Gen loss	-	Sum	-

We used the hugging face implementation of BERT-large and BART-large and optimized our models by AdamW (Loshchilov and Hutter, 2019). In addition, we use the two-layer fully connected layer with tanh as the intermediate activation function as the mapping function for the contextual representation. The dimension of the hidden layer is 1024. GELU (Hendrycks and Gimpel, 2016) also activates the output before being added to the other representations.

We first train the two models independently so that they can learn the contextual representation of event detection and event argument extraction tasks, respectively. Then in the joint bridging training stage, we also set different learning rates for different models. However, because their loss has decreased to the same magnitude in the process of independent training, the final learning rates obtained through grid search are the same size. We optimized the parameters with grid search, in the independent training: training epoch 40, learning rate $\in$ {1e-5, 2e-5, 1e-4}, training batch size with gradient accumulation $\in$ {1*8, 1*32, 1*64}, focal loss gamma $\in$ {1, 2, 3, 4, 5}, generation loss $\in$ {mean, sum}. As for the joint bridge training, we only used grid search to find the optimal learning rate $\in$ {5e-8, 5e-7, 5e-6, 2e-5} and fixed other parameters: training epoch 30, training batch size with gradient accumulation 1*32, focal loss gamma 3, generation loss "sum." Table 10 shows the final optimal parameter combination. Each experiment was conducted on NVIDIA GeForce RTX 3090 Core GPU 24GB. It is worth noting that the ERE-EN dataset has more noise than ACE 2005, so the model needs a larger learning rate on ERE-EN than that in ACE 2005. In other words, when trained independently on the ERE-EN dataset, the learning rates of BERT and BART are 3e-5 and 4e-5, respectively. In joint bridge training, the learning rates are 7e-7 and 8e-7, respectively.

While Inferring, our generative model generates sequences by greedy search, and the maximum sequence length is set according to dataset statistics, which is a bit larger than the length of the longest ground truth, for ACE05-E, ACE05-E$^+$, ERE-EN, its 50 tokens. As for the input length, the ACE05-E and ACE05-E$^+$ are 185 tokens, the ERE-EN is 325 tokens, and the detection model is consistent with the generative model. Besides, we parse the event records by template matching and slot mapping according to the ontology $\mathcal{O}$, as shown in Algorithm 1.

Reproduce Baselines. Among the baselines we selected, we tried to reproduce BERT_QA, BART-GEN, DEGREE, and GTEE-BASE. They all got similar results to those in the original paper report except for DEGREE and BERT_QA. Therefore, we used the experimental results reported in (Li et al., 2021b) and (Lin et al., 2020b) in Section 5.1, while the above two models used our experimental results. The hyperparameter Settings are listed as follows:

Table 11: Hyperparameter Settings for BART-GEN for our implementation and given in the original paper.

Parameter	Li et al. (2021b)	Our Implementation
Base model	BART-large	BART-large
Learning rate	[1e-5, 3e-5]	2e-5
Scheduler	Linear (without warmup)	Id.
Batch size	2*8	4*4
Max sequence length	512	512
Training epochs	[3,6]	20
Beam size	4	4

Algorithm 1 Extracting arguments from predicted template

Input: Predicted template $\mathcal{PT}$, Predefined template $\mathcal{GT}$, event type $\mathcal{E}$, Predefined Ontology $\mathcal{O}$ {e.g. $\mathcal{PT}$ is "Kelly met with officials in Seoul place", $\mathcal{GT}$ is "<arg1> met with <arg2> in <arg3> place"}

Initialization: predicted argument list $\mathcal{A}$, the pointers $p_{ptr} \leftarrow 0, t_{ptr} \leftarrow 0$
Split $\mathcal{PT}$ and $\mathcal{GT}$ into token lists, and ensure that <arg\d*> is a whole
while $p_{ptr} < |\mathcal{PT}|$ and $t_{ptr} < |\mathcal{GT}|$ **do**
  if $\mathcal{GT}\,[t_{ptr}]$ is <arg\d*> **then**
    $n \leftarrow$ \d*
    $role\_name = \mathcal{O}\,[\mathcal{E}]\,[n]$
  end if
  if $\mathcal{PT}\,[p_{ptr}]$ is <arg> **then**
    $p_{ptr} \leftarrow p_{ptr} + 1, t_{ptr} \leftarrow t_{ptr} + 1$
  else
    $arg_{start} = p_{ptr}, nxt_{ptr} = p_{ptr} + 1$
    while $nxt_{ptr} < |\mathcal{GT}|$ **do**
      if $\mathcal{GT}\,[nxt_{ptr}] ==$ <arg\d*> **then**
        Break
      end if
      $nxt_{ptr} \leftarrow nxt_{ptr} + 1$
    end while
    while $p_{ptr} < |\mathcal{PT}|$ **do**
      if $\mathcal{PT}\,[p_{ptr}] == \mathcal{GT}\,[t_{ptr}+1]$ and $\mathcal{PT}\,[p_{ptr} : p_{ptr} + nxt_{ptr} - t_{ptr} - 1] == \mathcal{GT}\,[t_{ptr}+1 : nxt_{ptr}]$ **then**
        Break
      end if
      $p_{ptr} \leftarrow p_{ptr} + 1$
    end while
    $\mathcal{A} \cup [\mathcal{E}, \mathcal{PT}\,[arg_{start} : p_{ptr}]\,, role\_name]$
  end if
end while

Output: $\mathcal{A}$

Table 12: Hyperparameter Settings for GTEE-BASE for our implementation and given in the original paper.

Parameter	Liu et al. (2022b)	Our Implementation
Base model	BART-large	BART-large
Learning rate	1e-5	2e-5
Batch size	32*8	4*8
Max sequence length	-	185 | 320
Max output length	-	78 | 100
Training epochs	40	40
Weight decay	1e-5	1e-5
Gradient clip	5.0	5.0
Warm-up ratio	10%	10%
Negative sample ratio	4%	3%

- **BART-GEN.** We follow the settings of the original paper when reproducing BART-GEN. However, there are some differences, and we list them in Table 11. In addition, since Li et al. (2021b) only implemented ACE05-E, we set the learning rate by referring to our model when implementing ACE05-E$^+$ and ERE, and other parameters were unchanged.

- **GTEE-BASE.** Since Liu et al. (2022b) have not open-sourced the code, we reproduced GTEE-BASE by ourselves with the hyperparameter settings shown in Table 12. Here, our negative sample ratio is $\frac{1}{evet\ type\ num}$, which means that ACE05-E and ACE05-E$^+$ are $\frac{1}{33}$ because they have 33 event types, and ERE-EN is $\frac{1}{38}$ because it has 38 event types. They all round off to a ratio of about 3%.

The hardware environment of these experiments is the same as that of HDGSE[3], as mentioned in the previous paragraph. On the other hand, the python environment is strictly set up according to the requirements of open-source codes.

Event Detection. In Section 5.1, we implement the classification-based and generation-based methods on the event detection task. As the classification-based approach, we used the event detection model of our HDGSE[3] in Section 3.2. Meanwhile, we use a template-based generative model for the generation-based approach, analogous to our event argument extraction model, but with templates and no prompts. We concatenate multiple templates for input sequences with multiple event mentions so that the model generates all event mentions sequentially. The template we used here was "Event type: <event> Trigger: <trg>." The experimental hyperparameters are consistent with those of main settings.

C Template Constructing

In this section, we show the three templates mentioned in Section 5. Their construction strategy is as follows:

- **No semantic integrity.** The first template is the least semantically complete; we take the role's name as a hint and add it before the argument slot, while the order of argument placement is random. For example, "Agent <arg1> Person <arg2> Place <arg3>". This template can only tell us the `Justice:Release-Parole` event's argument roles, but it does not form a natural sentence and has no semantic information.

- **Weak semantic integrity.** The second type of template maintains weak semantic integrity, and we use ontologies predetermined by Li et al. (2021c) as such templates, such as "<arg1> released or paroled <arg2> in <arg3> place". We can roughly understand that this is an `Justice:Release-Parole` event because the template mentions the two keywords "released or paroled." But this type of template misses the subject and role information, and the model may be confused. For example, "<arg1>" is not restricted to "Person" in the template, and the model may be likely to predict an "Institution" for it. Hence, the semantics of this kind of template is incomplete.

- **Strong semantic integrity.** The third template combines the advantages of the above two templates. It hints at the roles and ensures the sentence's semantic integrity. We refer to "APEX" defined by Wang et al. (2022), consider all roles to paraphrase each event, and arrange argument slots after each role, e.g., "an Entity <arg1> ends its custody of a Person <arg2> at a Place <arg3>". The sentence is semantically complete after removing slots.

ACE 2005 and ERE-EN are listed in Table 13 and Table 14, respectively.

Table 13: All templates of ACE05-E and ACE05-E$^+$ used in the main and ablation experiments in this paper.

Event	Type	Template
Movement:Transport	1	<arg1> transported <arg2> in <arg3> vehicle from <arg4> place to <arg5> place
	2	Agent <arg1> Artifact <arg2> Vehicle <arg3> Origin <arg4> Destination <arg5>
	3	an Agent <arg1> moves an Artifact <arg2> from Origin <arg4> to Destination <arg5> with Vehicle <arg3> at Price
Personnel:Elect	1	<arg1> elected <arg2> in <arg3> place
	2	Entity <arg1> Person <arg2> Place <arg3>
	3	a candidate Person <arg2> wins an election by voting Entity <arg1> at a Place <arg3>
Personnel:Start-Position	1	<arg1> started working at <arg2> organization in <arg3> place
	2	Person <arg1> Entity <arg2> Place <arg3>
	3	a Person <arg1> begins working for an Entity <arg2> or change office at a Place <arg3>
Personnel:Nominate	1	<arg1> nominated <arg2>
	2	Agent <arg1> Person <arg2>
	3	a Person <arg2> is nominated for a new position by another Agent <arg1> at a Place
Personnel:End-Position	1	<arg1> stopped working at <arg2> organization in <arg3> place
	2	Person <arg1> Entity <arg2> Place <arg3>
	3	a Person <arg1> stops working for an Entity <arg2> or change office at a Place <arg3>
Conflict:Attack	1	<arg1> attacked <arg2> hurting <arg5> victims using <arg3> instrument at <arg4> place
	2	Attacker <arg1> Target <arg2> Instrument <arg3> Place <arg4> Victim <arg5>
	3	An Attacker <arg1> physically attacks a Target <arg2> with Instrument <arg3> at a Place <arg4> hurting Victim <arg5>
Contact:Meet	1	<arg1> met with <arg2> in <arg3> place
	2	Entity <arg1> Entity <arg2> Place <arg3>
	3	one Entity <arg1> and another Entity <arg2> come together at same Place <arg3> and interact in person
Life:Marry	1	<arg1> married <arg2> in <arg3> place
	2	Person <arg1> Person <arg2> Place <arg3>
	3	one Person <arg1> and another Person <arg2> are married at a Place <arg3>
Transaction:Transfer-Money	1	<arg1> gave money to <arg2> for the benefit of <arg3> in <arg4> place
	2	Giver <arg1> Recipient <arg2> Beneficiary <arg3> Place <arg4>
	3	transfer Money from the Giver <arg1> to the Beneficiary <arg3> or Recipient <arg2> at a Place <arg4>
Conflict:Demonstrate	1	<arg1> demonstrated at <arg2> place
	2	Entity <arg1> Place <arg2>
	3	Entity <arg1> come together in a Place <arg2> to protest or demand official action
Business:End-Org	1	<arg1> organization shut down at <arg2> place
	2	Org <arg1> Place <arg2>
	3	an Organization Org <arg1> goes out of business at a Place <arg2>
Justice:Sue	1	<arg1> sued <arg2> before <arg3> court or judge in <arg4> place
	2	Plaintiff <arg1> Defendant <arg2> Adjudicator <arg3> Place <arg4>
	3	Plaintiff <arg1> initiate a court proceeding to determine the liability of a Defendant <arg2> judge by Adjudicator <arg3> at a Place <arg4>
Life:Injure	1	<arg1> injured <arg2> with <arg3> instrument in <arg4> place
	2	Agent <arg1> Victim <arg2> Instrument <arg3> Place <arg4>
	3	a Victim <arg2> experiences physical harm from Agent <arg1> with Instrument <arg3> at a Place <arg4>
Life:Die	1	<arg1> killed <arg2> with <arg3> instrument in <arg4> place
	2	Agent <arg1> Victim <arg2> Instrument <arg3> Place <arg4>
	3	life of a Victim <arg2> ends by an Agent <arg1> with Instrument <arg3> at a Place <arg4>
Justice:Arrest-Jail	1	<arg1> arrested <arg2> in <arg3> place
	2	Agent <arg1> Person <arg2> Place <arg3>
	3	the Agent <arg1> takes custody of a Person <arg2> at a Place <arg3>
Contact:Phone-Write	1	<arg1> communicated remotely with <arg2> at <arg3> place
	2	Entity <arg1> Entity <arg2> Place <arg3>
	3	phone or written communication between one Entity <arg1> and another Entity <arg2> at a Place <arg3>
Transaction:Transfer-Ownership	1	<arg1> gave <arg4> to <arg2> for the benefit of <arg3> at <arg5> place
	2	Seller <arg1> Buyer <arg2> Beneficiary <arg3> Artifact <arg4> Place <arg5>
	3	buying selling loaning borrowing giving receiving of Artifact <arg4> from Seller <arg1> to Buyer <arg2> or Beneficiary <arg3> at a Place <arg5> at Price
Business:Start-Org	1	<arg1> started <arg2> organization at <arg3> place
	2	Agent <arg1> Org <arg2> Place <arg3>
	3	an Agent <arg1> create a new Organization Org <arg2> at a Place <arg3>
Justice:Execute	1	<arg1> executed <arg2> at <arg3> place
	2	Agent <arg1> Person <arg2> Place <arg3>
	3	the life of a Person <arg2> is taken by an Agent <arg1> at a Place <arg3>
Justice:Trial-Hearing	1	<arg1> tried <arg2> before <arg3> court or judge in <arg4> place
	2	Prosecutor <arg1> Defendant <arg2> Adjudicator <arg3> Place <arg4>
	3	a court proceeding initiated to determine the guilty or innocence of the Defendant <arg2> Person with Prosecutor <arg1> and Adjudicator <arg3> at a Place <arg4>
Life:Be-Born	1	<arg1> was born in <arg2> place
	2	Person <arg1> Place <arg2>
	3	a Person <arg1> is born at a Place <arg2>
Justice:Charge-Indict	1	<arg1> charged or indicted <arg2> before <arg3> court or judge in <arg4> place
	2	Prosecutor <arg1> Defendant <arg2> Adjudicator <arg3> Place <arg4>
	3	a Defendant <arg2> is accused of a crime by a Prosecutor <arg1> for Adjudicator <arg3> at a Place <arg4>
Justice:Convict	1	<arg1> court or judge convicted <arg2> in <arg3> place
	2	Adjudicator <arg1> Defendant <arg2> Place <arg3>
	3	an Defendant <arg2> found guilty of a crime by Adjudicator <arg1> at a Place <arg3>
Justice:Sentence	1	<arg1> court or judge sentenced <arg2> in <arg3> place
	2	Adjudicator <arg1> Defendant <arg2> Place <arg3>
	3	the punishment for the Defendant <arg2> is issued by a state actor Adjudicator <arg1> at a Place <arg3>
Business:Declare-Bankruptcy	1	<arg1> declared bankruptcy at <arg2> place
	2	Org <arg1> Place <arg2>
	3	Organization Org <arg1> request legal protection from debt collection at a Place <arg2>
Justice:Release-Parole	1	<arg1> released or paroled <arg2> in <arg3> place
	2	Entity <arg1> Person <arg2> Place <arg3>
	3	an Entity <arg1> ends its custody of a Person <arg2> at a Place <arg3>
Justice:Fine	1	<arg1> court or judge fined <arg2> at <arg3> place
	2	Adjudicator <arg1> Entity <arg2> Place <arg3>
	3	a Adjudicator <arg1> issues a financial punishment Money to an Entity <arg2> at a Place <arg3>
Justice:Pardon	1	<arg1> court or judge pardoned <arg2> at <arg3> place
	2	Adjudicator <arg1> Defendant <arg2> Place <arg3>
	3	an Adjudicator <arg1> lifts a sentence of Defendant <arg2> at a Place <arg3>
Justice:Appeal	1	<arg1> appealed to <arg2> court or judge at <arg3> place
	2	Plaintiff <arg1> Adjudicator <arg2> Place <arg3>
	3	the decision for Defendant of a Plaintiff <arg1> is taken to a higher Place <arg3> for Adjudicator <arg2> review with Prosecutor
Justice:Extradite	1	<arg1> extradited <arg2> from <arg3> place to <arg4> place
	2	Agent <arg1> Person <arg2> Origin <arg3> Destination <arg4>
	3	a Person <arg2> is sent by an Agent <arg1> from Origin <arg3> to Destination <arg4>
Life:Divorce	1	<arg1> divorced <arg2> in <arg3> place
	2	Person <arg1> Person <arg2> Place <arg3>
	3	one Person <arg1> and another Person <arg2> are officially divorced at a Place <arg3>
Business:Merge-Org	1	<arg1> organization merged with <arg2> organization
	2	Org <arg1> Org <arg2>
	3	two or more Organizations Org <arg1> come together to form a new organization Org <arg2> at a Place
Justice:Acquit	1	<arg1> court or judge acquitted <arg2>
	2	Adjudicator <arg1> Defendant <arg2>
	3	a trial of Defendant <arg2> ends but Adjudicator <arg1> fails to produce a conviction at a Place

Table 14: All templates of ERE-EN used in the main and ablation experiments in this paper.

Event Type	Type	Template
Conflict:Attack	1	<arg1> attacked <arg2> using <arg3> instrument at <arg4> place
	2	Attacker <arg1> Target <arg2> Instrument <arg3> Place <arg4>
	3	An Attacker <arg1> physically attacks a Target <arg2> with Instrument <arg3> at a Place <arg4>
Justice:Acquit	1	<arg1> court or judge acquitted <arg2> at <arg3> place
	2	Adjudicator <arg1> Defendant <arg2> Place <arg3>
	3	a trial of Defendant <arg2> ends but Adjudicator <arg1> fails to produce a conviction at a Place <arg3>
Personnel:Elect	1	<arg1> elected <arg2> in <arg3> place
	2	Agent <arg1> Person <arg2> Place <arg3>
	3	a candidate Person <arg2> wins an election by voting Entity <arg1> at a Place <arg3>
Justice:Release-Parole	1	<arg1> released or paroled <arg2> in <arg3> place
	2	Agent <arg1> Person <arg2> Place <arg3>
	3	an Entity <arg1> ends its custody of a Person <arg2> at a Place <arg3>
Personnel:Nominate	1	<arg1> nominated <arg2> at <arg3> place
	2	Agent <arg1> Person <arg2> Place <arg3>
	3	a Person <arg2> is nominated for a new position by another Agent <arg1> at a Place <arg3>
Justice:Appeal	1	<arg1> appealed to <arg2> court or judge sentenced <arg3>
	2	Prosecutor <arg1> Adjudicator <arg2> Defendant <arg3>
	3	the decision for Defendant <arg3> of a Plaintiff is taken to a higher Place for Adjudicator <arg2> review with Prosecutor <arg1>
Transaction:Transfer-Ownership	1	<arg1> gave <arg4> to <arg2> for the benefit of <arg3> at <arg5> place
	2	Giver <arg1> Recipient <arg2> Beneficiary <arg3> Thing <arg4> Place <arg5>
	3	giving of Artifact Thing <arg4> from Giver <arg1> to Recipient <arg2> for the benefit of Beneficiary <arg3> at a Place <arg5>
Business:Declare-Bankruptcy	1	<arg1> declared bankruptcy
	2	Org <arg1>
	3	Organization Org <arg1> request legal protection from debt collection
Contact:Meet	1	<arg1> met face-to-face with <arg2> in <arg3> place
	2	Entity <arg1> Entity <arg2> Place <arg3>
	3	one Entity <arg1> and another Entity <arg2> come together at same Place <arg3> and interact in person
Life:Marry	1	<arg1> married <arg2> in <arg3> place
	2	Person <arg1> Person <arg2> Place <arg3>
	3	one Person <arg1> and another Person <arg2> are married at a Place <arg3>
Life:Divorce	1	<arg1> divorced <arg2> in <arg3> place
	2	Person <arg1> Person <arg2> Place <arg3>
	3	one Person <arg1> and another Person <arg2> are officially divorced at a Place <arg3>
Business:Merge-Org	1	<arg1> organization merged with <arg2> organization
	2	Org <arg1> Org <arg2>
	3	two or more Organizations Org <arg1> come together to form a new organization Org <arg2> at a Place
Contact:Correspondence	1	<arg1> communicated remotely with <arg2> at <arg3> place
	2	Entity <arg1> Entity <arg2> Place <arg3>
	3	one Entity <arg1> communicated remotely with another Entity <arg2> at a Place <arg3>
Contact:Contact	1	<arg1> communicated with <arg2> at <arg3> place
	2	Entity <arg1> Entity <arg2> Place <arg3>
	3	one Entity <arg1> communicated with another Entity <arg2> face to face at a Place <arg3>
Manufacture:Artifact	1	<arg1> manufactured or created or produced <arg2> at <arg3> place
	2	Agent <arg1> Artifact <arg2> Place <arg3>
	3	an Agent <arg1> manufactured or created or produced Artifact <arg2> at a Place <arg3>
Movement:Transport-Person	1	<arg1> transported <arg2> in <arg3> instrument from <arg4> place to <arg5> place
	2	Agent <arg1> Person <arg2> Instrument <arg3> Origin <arg4> Destination <arg5>
	3	an Agent <arg1> transported a Person <arg2> in Instrument <arg3> from Origin <arg4> place to Destination <arg5>
Movement:Transport-Artifact	1	<arg1> transported <arg2> from <arg3> place to <arg4> place
	2	Agent <arg1> Artifact <arg2> Origin <arg3> Destination <arg4>
	3	an Agent <arg1> transported Artifact <arg2> from Origin <arg3> place to Destination <arg4>
Contact:Broadcast	1	<arg1> communicated to <arg2> at <arg3> place (one-way communication)
	2	Entity <arg1> Audience <arg2> Place <arg3>
	3	an Entity <arg1> one-way communicated to one or more Audience <arg2> at a Place <arg3>
Transaction:Transaction	1	<arg1> gave something to <arg2> for the benefit of <arg3> at <arg4> place
	2	Giver <arg1> Recipient <arg2> Beneficiary <arg3> Place <arg4>
	3	a Giver <arg1> gave something to a Recipient <arg2> for the benefit of Beneficiary <arg3> at a Place <arg4>
Personnel:Start-Position	1	<arg1> started working at <arg2> organization in <arg3> place
	2	Person <arg1> Entity <arg2> Place <arg3>
	3	a Person <arg1> begins working for an Entity <arg2> or change office at a Place <arg3>
Justice:Pardon	1	<arg1> court or judge pardoned <arg2> at <arg3> place
	2	Adjudicator <arg1> Defendant <arg2> Place <arg3>
	3	an Adjudicator <arg1> lifts a sentence of Defendant <arg2> at a Place <arg3>
Justice:Fine	1	<arg1> court or judge fined <arg2> at <arg3> place
	2	Adjudicator <arg1> Entity <arg2> Place <arg3>
	3	a Adjudicator <arg1> issues a financial punishment Money to an Entity <arg2> at a Place <arg3>
Justice:Trial-Hearing	1	<arg1> tried <arg2> before <arg3> court or judge in <arg4> place
	2	Prosecutor <arg1> Defendant <arg2> Adjudicator <arg3> Place <arg4>
	3	a court proceeding initiated to determine the guilty or innocence of the Defendant <arg2> Person with Prosecutor <arg1> and Adjudicator <arg3> at a Place <arg4>
Business:End-Org	1	<arg1> organization shut down at <arg2> place
	2	Org <arg1> Place <arg2>
	3	an Organization Org <arg1> goes out of business at a Place <arg2>
Justice:Sue	1	<arg1> sued <arg2> before <arg3> court or judge in <arg4> place
	2	Plaintiff <arg1> Defendant <arg2> Adjudicator <arg3> Place <arg4>
	3	Plaintiff <arg1> initiate a court proceeding to determine the liability of a Defendant <arg2> judge by Adjudicator <arg3> at a Place <arg4>
Life:Injure	1	<arg1> injured <arg2> with <arg3> instrument in <arg4> place
	2	Agent <arg1> Victim <arg2> Instrument <arg3> Place <arg4>
	3	a Victim <arg2> experiences physical harm from Agent <arg1> with Instrument <arg3> at a Place <arg4>
Justice:Arrest-Jail	1	<arg1> arrested <arg2> in <arg3> place
	2	Agent <arg1> Person <arg2> Place <arg3>
	3	the Agent <arg1> takes custody of a Person <arg2> at a Place <arg3>
Justice:Execute	1	<arg1> executed <arg2> at <arg3> place
	2	Agent <arg1> Person <arg2> Place <arg3>
	3	the life of a Person <arg2> is taken by an Agent <arg1> at a Place <arg3>
Conflict:Demonstrate	1	<arg1> demonstrated at <arg2> place
	2	Entity <arg1> Place <arg2>
	3	Entity <arg1> come together in a Place <arg2> to protest or demand official action
Justice:Sentence	1	<arg1> court or judge sentenced <arg2> in <arg3> place
	2	Adjudicator <arg1> Defendant <arg2> Place <arg3>
	3	the punishment for the Defendant <arg2> is issued by a state actor Adjudicator <arg1> at a Place <arg3>
Life:Die	1	<arg1> killed <arg2> with <arg3> instrument in <arg4> place
	2	Agent <arg1> Victim <arg2> Instrument <arg3> Place <arg4>

(continued from previous page.)

	3	life of a Victim <arg2> ends by an Agent <arg1> with Instrument <arg3> at a Place <arg4>
Business:Start-Org	1	<arg1> started <arg2> organization at <arg3> place
	2	Agent <arg1> Org <arg2> Place <arg3>
	3	an Agent <arg1> create a new Organization Org <arg2> at a Place <arg3>
Personnel:End-Position	1	<arg1> stopped working at <arg2> organization in <arg3> place
	2	Person <arg1> Entity <arg2> Place <arg3>
	3	a Person <arg1> stops working for an Entity <arg2> or change office at a Place <arg3>
Justice:Extradite	1	<arg1> extradited <arg2> from <arg3> place to <arg4> place
	2	Agent <arg1> Person <arg2> Origin <arg3> Destination <arg4>
	3	a Person <arg2> is sent by an Agent <arg1> from Origin <arg3> to Destination <arg4>
Justice:Charge-Indict	1	<arg1> charged or indicted <arg2> before <arg3> court or judge in <arg4> place
	2	Prosecutor <arg1> Defendant <arg2> Adjudicator <arg3> Place <arg4>
	3	a Defendant <arg2> is accused of a crime by a Prosecutor <arg1> for Adjudicator <arg3> at a Place <arg4>
Transaction:Transfer-Money	1	<arg1> gave money to <arg2> for the benefit of <arg3> in <arg4> place
	2	Giver <arg1> Recipient <arg2> Beneficiary <arg3> Place <arg4>
	3	transfer Money from the Giver <arg1> to the Beneficiary <arg3> or Recipient <arg2> at a Place <arg4>
Justice:Convict	1	<arg1> court or judge convicted <arg2> in <arg3> place
	2	Adjudicator <arg1> Defendant <arg2> Place <arg3>
	3	an Defendant <arg2> found guilty of a crime by Adjudicator <arg1> at a Place <arg3>
Life:Be-Born	1	<arg1> was born in <arg2> place
	2	Person <arg1> Place <arg2>
	3	a Person <arg1> is born at a Place <arg2>