What Is Overlap Knowledge in Event Argument Extraction? APE: A Cross-datasets Transfer Learning Model for EAE

Kaihang Zhang[1], **Kai Shuang** [1]*, **Xinyue Yang**[1], **Xuyang Yao**[2] and **Jinyu Guo**[13]

[1]State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications
[2]China Telecom Research Institute
[3]University of Cambridge
{zkh1999, shuangk, crescent3919, guojinyu}@bupt.edu.cn
yaoxy11@chinatelecom.cn

Abstract

The EAE task extracts a structured event record from an event text. Most existing approaches train the EAE model on each dataset independently and ignore the overlap knowledge across datasets. However, insufficient event records in a single dataset often prevent the existing model from achieving better performance. In this paper, we clearly define the overlap knowledge across datasets and split the knowledge of the EAE task into overlap knowledge across datasets and specific knowledge of the target dataset. We propose APE model to learn the two parts of knowledge in two serial learning phases without causing catastrophic forgetting. In addition, we formulate both learning phases as conditional generation tasks and design Stressing Entity Type Prompt to close the gap between the two phases. The experiments show APE achieves new state-of-the-art with a large margin in the EAE task. When only ten records are available in the target dataset, our model dramatically outperforms the baseline model with average 27.27% F1 gain.[1]

1 Introduction

Event extraction (EE) is a pivotal task in information extraction. Typically, the event extraction task can be divided into two sub-tasks: event detection (ED) and event argument extraction (EAE). Thanks to recent works (Liu et al., 2022a; Sheng et al., 2022; Lai et al., 2020), event detection has achieved significant progress. The main challenge of EE lies in the EAE task.

The EAE task aims to extract a structured event record from an event text. Since different datasets often have various event types and argument structures, most studies (Ma et al., 2022; Lu et al., 2021; Liu et al., 2022b) train the EAE model on each dataset independently, such as ACE 2005 (Doddington et al., 2004), RAMS (Ebner et al., 2020), and WikiEvents (Li et al., 2021). However, one single dataset often cannot provide sufficient event records, which seriously prevents those models from achieving better performance. Especially in some industrial applications, the in-domain event record collection incurs expensive and time-consuming manual annotation. We argue that there is abundant transferable all-purpose knowledge of the EAE task among different datasets, called overlap knowledge. Exploring the overlap knowledge from existing datasets can significantly improve the model's performance and reduce the need for newly annotated data.

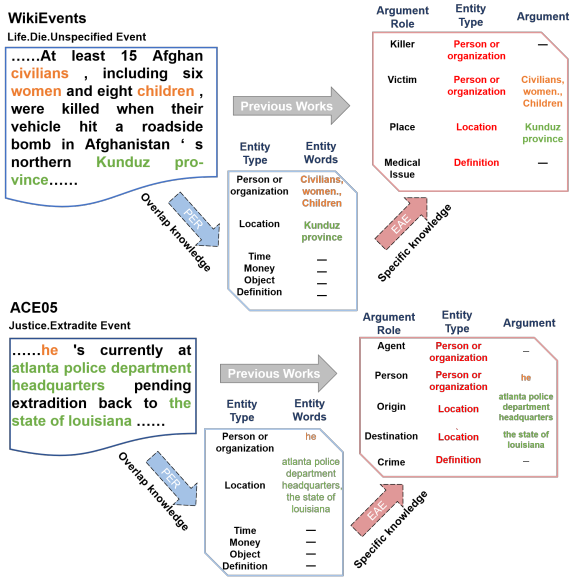

Figure 1: The illustration of overlap knowledge

How to transfer knowledge across datasets has yet to be well studied. Only Zhou et al. (2022) attempted to introduce variational information bottleneck to retain the shared knowledge between two datasets and achieved considerable success. Nevertheless, their model architecture restricts that they can only obtain overlap knowledge from up to two datasets. Moreover, it has not explicitly defined *what is the overlap knowledge among the different*

*Corresponding author.
[1]https://github.com/ZKH-1999/APE

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 393–409
July 9-14, 2023

datasets. Therefore, they use the EAE task's training objective to train the model on two datasets jointly and roughly let the model distinguish what knowledge is shareable across datasets. The imprecise training objectives perplex the model to learn the overlap knowledge better.

In this work, we propose a Seek Common ground while Reserving Differences (SC-RD) framework to define the overlap knowledge clearly. SC-RD suggests defining overlap knowledge based on a cross-dataset common ground and isolating other knowledge into specific knowledge. As shown in Figure 1, every argument role in different datasets can be attached to an entity type. We introduce a finite entity type set (shown in Appendix Table 6) as the common ground across datasets. Based on the entity type set, we define the overlap knowledge as *identifying entity words associated with the event by a given entity type*. The specific knowledge is defined as *identifying arguments based on the output of overlap knowledge*. As illustrated in Figure 1, the two knowledge split the EAE task into two steps: In the first step, the model uses the overlap knowledge to focus on the entity word associated with the event. The second step finishes the EAE task based on the specific knowledge. Therefore, the EAE task can be reformulated as the product of two conditional probabilities:

$$p\left(\mathcal{A}|\mathcal{X},\ K\right) \propto p\left(w|\mathcal{X}, k_o\right) p\left(\mathcal{A}|w, \mathcal{X}, k_s\right) \quad (1)$$

where $\mathcal{A}$ is the event argument, w are event-related entity words, and $\mathcal{X}$ donates the event text. $k_o \in K$ represents overlap knowledge, and $k_s \in K$ represents specific knowledge. $p\left(w|\mathcal{X}, k_o\right)$ is independent of datasets and can be learned from a pseudo-entity recognition (PER) task on multi-datasets straightforwardly. The PER only identifies the entity words associated with the event so that EAE labels can be converted to PER labels by a manual mapping function. The structure definition of $\mathcal{A}$ varies with the dataset, so we learn $p\left(\mathcal{A}|w, \mathcal{X}, k_s\right)$ from the EAE task on the target dataset based on the overlap knowledge.

We implement the above idea in APE, which **A**ssembles two **P**arameter-**E**fficient tuning methods to harmonize two parts of knowledge in one single model. Specifically, we introduce two learning phases (illustration in Figure 2) to learn overlap and specific knowledge, respectively. In the overlap learning phase, we merge multi-datasets and convert their unaligned EAE labels to aligned PER labels to optimize the Prefix, which is introduced to save overlap knowledge. In the specific learning phase, we load and freeze the trained Prefix and tune the Adapter's parameters with the EAE task in the target dataset to save specific knowledge. All the pre-trained model's parameters will be frozen like traditional parameter-efficient tuning methods. Furthermore, to ensure the overlap knowledge plays a part in the EAE task, we format both training tasks as conditional generation tasks and propose the Stressing Entity Type Prompt to ignite the overlap knowledge in the EAE task.

To the best of our knowledge, we are the first to clearly define the overlap knowledge across datasets, so we can give the model a transparent training objective to help it learn the overlap knowledge. Our model expands parameter-efficient tuning methods to the transfer learning scene. Since APE optimizes different parameters in two learning phases, learning the specific knowledge will not trigger catastrophic forgetting (McCloskey and Cohen, 1989) of the overlap knowledge.

We have conducted extensive experiments on three widely used datasets. The experimental results show that our proposed APE outperforms baselines with a large margin (2.7%, 2.1%, 3.4% F1 gain absolutely on three benchmarks). Moreover, it achieves 27.27% F1 score gain average over three datasets when only ten samples of the target dataset are available, indicating our model's few-shot learning ability. Further analysis in Section 4.3 confirms the efficacy of the main components in our model.

2 Method

As illustrated in Figure 2, APE learns two parts of knowledge in two learning phases sequentially. To overcome catastrophic forgetting, our model (Section 2.2) assembles Prefix (Li and Liang, 2021) to save overlap knowledge and Adapter (Houlsby et al., 2019) to save specific knowledge, respectively. To fully use the overlap knowledge learned from multi-datasets, we carefully design the Task Formulation (Section 2.1) and the Stressing Entity Type Prompt (Section 2.3) of two learning phases.

2.1 Task Formulation

Our approach introduces PER task to learn overlap knowledge and EAE task to learn specific knowledge. Every NLP task can be treated as a "text-to-text" problem (Raffel et al., 2020). Our approach

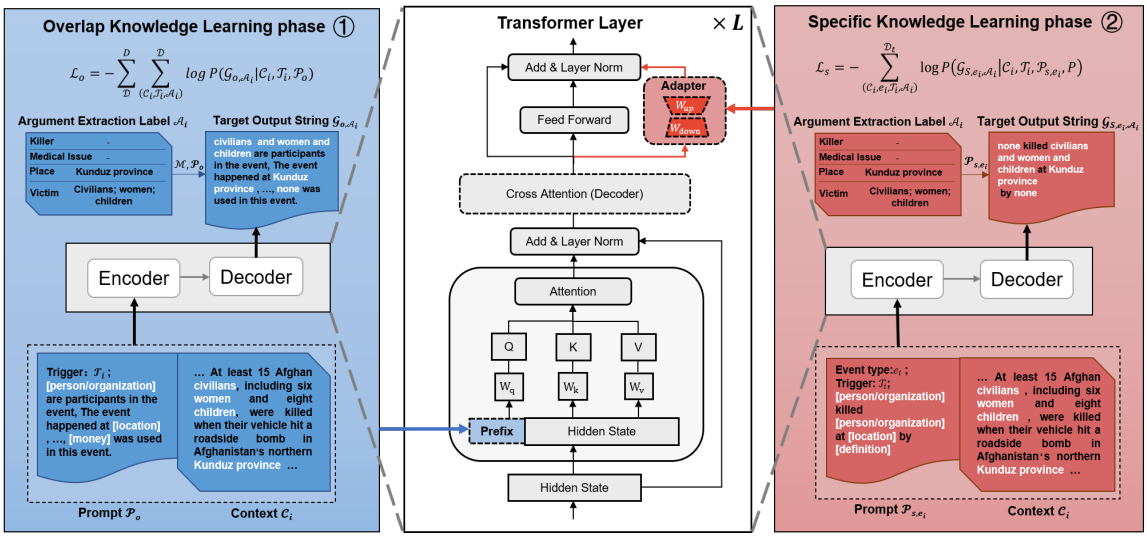

Figure 2: The framework of our APE model

formats both learning phases as conditional generation problems to narrow the gap between the two learning phases.

Specifically, we define the event dataset as $\mathcal{D} = \{(\mathcal{C}_i, e_i, \mathcal{T}_i, \mathcal{A}_i)|i < |\mathcal{D}|\}$, where C_i is the ith event context. e_i and $\mathcal{T}_i$ are the event type and trigger of the ith event separately. $\mathcal{A}_i = \{(r_j, span_j), \dots\}$ is the argument set of the event, where r_j denotes the argument role, and $span_j$ is the offset of the argument. For both phases, the input of our model is a designed prompt $\mathcal{P}$ and a context $\mathcal{C}_i$. The target output string is an answered prompt $\mathcal{G}$ containing the answer to the task. The language model (LM) models the conditional probability of answered prompt $\mathcal{G}$ as:

$$p\left(\mathcal{G}|\mathcal{X},\ \theta\right) = \prod_{i=1}^{|\mathcal{G}|} p\left(g_i|g_{<i}, \mathcal{X}\right) \tag{2}$$

$$\mathcal{X} = \left[\mathcal{P}; \left[SEP\right]; \mathcal{C}_i\right] \tag{3}$$

Where $\mathcal{X}$ is the input of the model, θ donates the parameters of LM. The construction of $\mathcal{P}$ and $\mathcal{G}$ in two learning phases will be respectively described in section 2.3.

2.2 Model Architecture

Our APE model assembles Prefix and Adapter into pre-trained encoder-decoder Transformer (Vaswani et al., 2017). The model can acquire two parts of knowledge without causing catastrophic forgetting by optimizing different parameter regions in two learning phases.

For overlap knowledge, we equip each self-attention module with a short Prefix vector $P \in \mathbb{R}^{|P| \times d_{model}}$ to represent and save it. In each layer, the new self-attention module with overlap knowledge intervention is formalized as:

$$H \leftarrow LayerNorm\left(H^{'} + H\right) \tag{4}$$

$$H' = MHSA\left(P \oplus H\right)_{|P|:|P \oplus H|} \tag{5}$$

Where $MHSA(\bullet)$ denotes the multi-head self-attention mechanism, and $(\bullet)_{a:b}$ donates the slicing operation on the seq_len dim from a to b. The Prefix will be assembled into the model in both learning phases since we use the overlap knowledge in the specific knowledge learning phase too. We optimize the Prefix P only in the overlap knowledge learning phase, and freeze it in the specific knowledge learning phase.

For specific knowledge, we adopt an Adapter parallel with the feed-forward module to represent and save it. The Adaptor locates behind the Prefix to model the order of knowledge utilization in the SC-RD framework. The specific knowledge will be involved in the computation of H_{ad}, and the new feed-forward module with Adapter is formalized as:

$$H \leftarrow LayerNorm\left(H + H_{ffd} + H_{ad}\right) \tag{6}$$

$$H_{ad} = W_{up}\ \sigma\left(W_{down} H\right) \tag{7}$$

Where $W_{down} \in \mathbb{R}^{d_{model} \times d_{adapter}}$ and $W_{up} \in \mathbb{R}^{d_{adapter} \times d_{model}}$ are tunable parameters in the Adapter, $\sigma(\bullet)$ is a nonlinear activation function, and H_{ffd} represents the output of the feed-forward layer. Only in the specific knowledge learning phase, we assemble the Adapter into the model

and optimize it. Like traditional parameter-efficient tuning methods, the pre-trained parameters of the Transformer are frozen in both phases.

2.3 Stressing Entity Type Prompt

The Stressing Entity Type Prompt can indicate the model to generate words with the corresponding entity type in the designated location. We design the prompts under the same style in two learning phases, which uses identical special tokens to mark entity types. In the EAE task, those special tokens will ignite the overlap knowledge.

2.3.1 Overlap Knowledge Learning Phase

We introduce the PER task to align the diverse datasets and learn overlap knowledge from them. To convert EAE labels to PER labels, we manually create a mapping function $\mathcal{M}(r)$ which maps each argument role r to an entity type.

Prompt Construction The overlap knowledge is independent of datasets, so all datasets' prompt in the overlap knowledge learning phase is identical. Entity-type special tokens mark the position expected to be filled by the model and the corresponding entity types. The model should recognize the right entity words by referring to the context $\mathcal{C}_i$. The manual overlap knowledge prompt $\mathcal{P}_o$ was designed as:

> *[person/organization]* are a participant in the event, the event happened at *[location]*, *[object]* are relate to the event, *[definition]* are the terminology in the event, the event taken place at *[time]*, *[money]* was used in this event.

[•] represents an entity-type special token, and the prompt natively contains the congruent relationship between the special token and entity type. Furthermore, we concatenate the event trigger $\mathcal{T}_i$ of the given event with the prompt to help the model focus on the correct event.

Target Output String Construction As shown in Figure 2 ①, for an event context $\mathcal{C}_i$ and its arguments $\mathcal{A}_i$ sampled from any event dataset, we first convert $\mathcal{A}_i$ to the PER label according to $\mathcal{M}$. Then, we construct the ground truth generation sequence $\mathcal{G}_{o,\mathcal{A}_i}$ by filling the PER label into $\mathcal{P}_o$. If several words are categorized as the same type, they will be concatenated by "and". If there is an empty set of some entity types, we fill "none" into $\mathcal{P}_o$ to replace the special token.

2.3.2 Specific Knowledge Learning Phase

We learn the specific knowledge by finishing the EAE task based on the overlap knowledge. To ignite the overlap knowledge contained in the Prefix, we inherit entity-type special tokens from the overlap knowledge learning phase and build prompts according to the target dataset with those special tokens.

Prompt Construction In the target dataset, for each event type e_i, we refer pre-defined prompt from Ma et al. (2022) and replace the textual argument roles in the prompt with the above entity-type special token according to $\mathcal{M}$. The entity-type special token hints to the model what entity type of words are most likely to serve as this argument role. For example, given an event type e: *Life.Die.Unspecified*, the renovated prompt $\mathcal{P}_{s,e}$ can be got as:

> **Prompt from Ma et al. (2022):**
> *Killer* killed *Victim* at *Place* by *MedicalIssue*
> **Renovated prompt:**
> *[person/organization]* killed *[person/organization]* at *[location]* by *[definition]*

As shown in Figure 2 ②, following Ma et al. (2022), we concatenate the event type e_i and the event trigger $\mathcal{T}_i$ of the given event sample with the renovated prompt.

Target Output String Construction For each event record $(\mathcal{C}_i, e_i, \mathcal{T}_i, \mathcal{A}_i)$ sampled from the target dataset, as shown in Figure 2 ②, we construct the ground truth generation sequence $\mathcal{G}_{s,e_i,\mathcal{A}_i}$ by filling $\mathcal{A}_i$ into $\mathcal{P}_{s,e_i}$. Like the overlap knowledge learning phase, arguments with the same role will be concatenated by "and" and the uninvolved argument role will be filled by "none".

2.4 Training, Inference, and Decoding

Training First, in the overlap knowledge learning phase, the trainable parameters of our model are only the Prefix P in each layer and the embedding of entity-type special tokens. The Adapter will be disabled. The training objective is to maximize $p\left(w|\mathcal{X}, k_o\right)$ of Equation 1, which is equivalent to minimizing the negative loglikelihood loss in multi-datasets $D = \{\mathcal{D}_1, \mathcal{D}_2 \ldots\}$:

$$\mathcal{L}_o = -\sum_{\mathcal{D}}^{D} \sum_{(\mathcal{C}_i, \mathcal{T}_i, \mathcal{A}_i)}^{\mathcal{D}} \log\left(P\left(\mathcal{G}_{o,\mathcal{A}_i}|\mathcal{C}_i, \mathcal{T}_i, \mathcal{P}_o\right)\right) \tag{8}$$

Then, in the specific knowledge learning phase, we load and freeze all parameters learned from the overlap knowledge learning phase and assemble the Adapter into our model to save the specific knowledge. Only W_{down} and W_{up} in the Adapter will be optimized. The training objective is to maximize $p\left(\mathcal{A}|w, \mathcal{X}, k_s\right)$ by minimizing the negative loglikelihood in the target dataset $\mathcal{D}_t$:

$$\mathcal{L}_s = -\sum_{(\mathcal{C}_i, e_i, \mathcal{T}_i, \mathcal{A}_i)}^{\mathcal{D}_t} \log\left(P\left(\mathcal{G}_{S,e_i,\mathcal{A}_i}|\mathcal{C}_i, \mathcal{T}_i, \mathcal{P}_{s,e_i}, P\right)\right) \tag{9}$$

Where P is the Prefix.

Inference In the inference stage, we assemble the trained Prefix and Adapter into the model. The input of APE is as same as the specific learning phase. Our model generates sequence by beam search strategy with $width = 10$. The maximum sequence length is set to 100 tokens, which is plenty for every dataset.

Decoding Routinely, we decode the arguments from generated sequence by using regular expressions according to the $\mathcal{P}_{s,e}$ for each sample. It is rare, but not all generated sequences are valid. For the argument roles we cannot decode from the generated sequence, we set "none" to them. Following Lu et al. (2021), we obtain the offset of the argument by finding the nearest matched string to the event trigger $\mathcal{T}_i$.

3 Experiments Setup

3.1 Datasets

We evaluate our model on three popular datasets: ACE 2005 (Doddington et al., 2004), RAMS (Ebner et al., 2020), and WikiEvents (Li et al., 2021). ACE05 is a classical sentence-level dataset. We follow Wadden et al. (2019)'s pre-processing scripts on ACE05. RAMS and WikiEvents are both document-level datasets. Since the context of the document-level dataset sometimes exceeds the constraint, we follow Ma et al. (2022), which adds a window centering on the trigger words and only encodes the words within the window. The statistics of the datasets are listed in Appendix Table 7. The multi-datasets $D = \{ACE05,\ RAMS,\ WikiEvents\}$ in this work.

3.2 Baselines

We compare our APE model with the following state-of-the-art baseline models: (1) OneIE (Lin et al., 2020) jointly extracts the globally optimal IE result from a context. (2) EEQA (Du and Cardie, 2020) regards the event argument extraction task as an end-to-end question-answering (QA) task. (3) BART-Gen (Li et al., 2021) proposes a conditional generation approach to complete document-level EAE task. (4) PAIE (Ma et al., 2022) utilizes multi-role prompts under extractive settings to capture argument interactions. (5) PAIE-Joint uses the same model in PAIE, but joint train the model in three datasets for a fair comparison with our model. (6) UnifiedEAE (Zhou et al., 2022) introduces variational information bottleneck to explore shared knowledge from two EAE datasets.

3.3 Evaluation Metric

Following baseline models, we adopt two metrics: Arg-I and Arg-C. Following Li et al. (2021), we add Head-C for WikiEvents datasets. Please refer to Appendix A for the detail of evaluation metric.

3.4 Implementation Details

We initialize the weight of the Transformer with BART model (Lewis et al., 2020). The length $|P|$ of Prefix is set to 70, and the inter-dim $d_{adapter}$ of the Adapter is set to 512 for BART-base model and 768 for BART-large model. For simplicity, we initialize the Prefix and the Adapter randomly. We optimized our models on NVIDIA A40 GPU by AdamW (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$, and 10% warmup steps. We set the learning rate to 1e-3 for Prefix and 1e-4 for Adapter. To ensure the confidence of the result, we repeated the model training five times with five fixed seeds [14, 21, 28, 35, 42]. The reported experimental results are the average score. We exhibit some examples of $\mathcal{M}(r)$ (Table 10) and prompts (Table 11) in the Appendix. The complete $\mathcal{M}(r)$ and prompts of each dataset are available in our codebase.

4 Results and Analyses

To investigate the efficacy of our APE model, we compare our model with several state-of-the-art baseline models (4.1). Then, we verify the significance of transfer overlap knowledge (4.2) in the few-shot setting. We also perform ablation studies and further analysis to examine the effectiveness of the main components in our model (4.3).

4.1 Overall Performance

Table 1 present the main result of all baseline models and APE on three datasets. APE refers to our

Table 1: The Overall performance of our model and baselines. We **bold** the best result and underline the second best. b in column PLM denotes base model and l is large model.

Model	PLM	ACE05		RAMS		WikiEvents		
		Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Head-C
OneIE	BERT-b	65.9	59.2	-	-	-	-	-
	BERT-l	73.2	69.2	-	-	-	-	-
EEQA	BERT-b	68.2	65.4	46.4	44.0	54.3	53.2	56.9
	BERT-l	70.5	68.9	48.7	46.7	56.9	54.5	59.3
BART-Gen	BART-b	59.6	55.0	50.9	44.9	47.5	41.7	44.2
	BART-l	69.9	66.7	51.2	47.1	66.8	62.4	65.4
PAIE	BART-b	73.6	69.8	54.7	49.5	68.9	63.4	66.5
	BART-l	75.7	72.7	56.8	52.2	70.5	65.3	68.4
PAIE-Joint	BART-b	73.8	69.5	53.3	48.3	69.3	63.7	65.9
	BART-l	75.1	72.4	55.9	51.8	70.1	65.2	67.9
UnifiedEAE	BART-b	76.1	71.9	55.5	49.9	69.8	64.0	66.3
APE(Single)	BART-b	74.1	70.1	54.8	49.6	66.2	62.1	64.9
	BART-l	75.3	72.9	56.3	51.7	70.6	65.8	68.4
APE	BART-b	75.5	72.9	56.1	51.6	70.7	66.0	68.7
	BART-l	**78.2**	**75.4**	**58.1**	**54.3**	**73.7**	**68.7**	**70.8**

full model, which optimizes the Prefix in multi-datasets. APE(Single) refers to the APE model trained in the transfer-disable setting, which optimizes the Prefix only in the target dataset. In the APE(Single), the overlap knowledge degrades into shared knowledge between different event types within the same target dataset.

From Table 1, we have the following observations. First, APE achieves the highest F1 score on every evaluation metric compared with all the baselines model. Our base model obtained +1%, +1.7%, and +2% gain of Arg-C F1 scores on ACE05, RAMS, and WikiEvents, respectively. The large model expands the margin to +2.7%, +2.1%, and +3.4%. The results show that there is abundant overlap knowledge in multi-datasets, and our model can fully utilize it in the target dataset. Second, despite not relying on transfer learning, APE (Single) also achieves state-of-the-art performance on ACE05 and WikiEvents, and a competitive score on RAMS, which suggests that knowledge shared between different event types in a single dataset can also boost performance. Third, the PAIE-Joint even slightly worse than the PAIE. It donate that it is difficult for the model to find overlap knowledge by itself from datasets with various event structures, event types, and even different annotation guidelines. The APE can exploit the overlap knowledge from the transparent training objective of the PER task, and achieve better performance.

Table 2: Arg-C F1 score on few-shot setting

Dataset		ACE05	RAMS	Wiki.
PAIE	10	$3.3_{\pm 2.1}$	$4.3_{\pm 1.4}$	$5.7_{\pm 3.6}$
	50	$35.2_{\pm 5.3}$	$25.2_{\pm 6.1}$	$31.4_{\pm 4.6}$
	100	$39.6_{\pm 2.5}$	$30.4_{\pm 2.1}$	$42.1_{\pm 3.2}$
	200	$51.2_{\pm 1.3}$	$35.8_{\pm 1.9}$	$53.2_{\pm 1.7}$
APE	10	$32.1_{\pm 7.1}$	$26.3_{\pm 4.2}$	$36.7_{\pm 8.3}$
	50	$42.5_{\pm 3.9}$	$33.4_{\pm 4.1}$	$47.6_{\pm 5.4}$
	100	$53.2_{\pm 1.7}$	$38.5_{\pm 1.6}$	$55.6_{\pm 2.6}$
	200	$59.3_{\pm 0.9}$	$41.1_{\pm 1.2}$	$59.5_{\pm 1.5}$

4.2 Few-shot Setting

APE is exceptionally suited for lacking in-domain labeled data because APE can learn from out-domain event records. Therefore, we conduct a few-shot experiment to verify the ability of APE to reduce the dependence on target dataset samples. Specifically, we optimize Prefix on the other two intact datasets and train Adapter on the target dataset with few samples.

Table 2 reports the Arg-C F1 score in the target dataset with 10, 50, 100, and 200 random sampled event records. From the results, we obtain the following observations. 1). APE significantly outperforms the state-of-the-art baseline PAIE model in three benchmarks. 2). Especially in the case of only ten samples, APE achieves 27.27% F1 score gains average in three datasets. 3). APE with 200 samples achieves competitive scores with some

Table 3: The performance of different variants on ACE05

Variant	Param		ACE05
	overlap	specific	
APE	Prefix	Adapter	72.9
$\text{APE}_{reversed}$	Adapter	Perfix	72.1
w/o Prefix	BART	Adapter	71.5
w/o Adapter	Prefix	BART	71.7
BART	BART	BART	69.4

baseline model trained on the whole WikiEvents or ACE05 dataset. The results indicate that APE significantly reduces the need for the scale of the target dataset.

4.3 Detailed Analysis

In this section, we study the effectiveness of the main components in our model and take a deeper look at what contributes to APE's final performance. All experiments will be based on the base-version model and report the average Arg-C F1 scores on five seeds. The experimental conclusions are also proper for the large version model.

4.3.1 Model Architecture Design

We first explore the effectiveness of APE model architecture in preventing catastrophic forgetting. We tried variants of APE as follows: 1) $\text{APE}_{reversed}$: it has the same model architecture as APE but saves overlap knowledge in the Adapter and specific knowledge in the Prefix. 2) w/o Prefix: it is an APE without Prefix, which updates all pre-trained parameters to save overlap knowledge. 3) w/o Adapter: pre-trained parameters will be updated to save specific knowledge. 4) BART: it is a standard BART model without additional parameters. We optimize the model in the overlap knowledge learning phase and fine-tune it in the specific knowledge learning phase.

The result of ACE05 is summarized in Table 3, and the result of other datasets is in Appendix Table 8. All variants that save overlap and specific knowledge into different parameters outperform the plain BART model significantly. Since the plain BART model saves overlap and specific knowledge in the same parameters, serial learning phases will lead to catastrophic forgetting of previous knowledge.

Suppose we save both knowledges into new parameter regions (APE, $\text{APE}_{reversed}$). In this case, we can also obtain a considerable performance gain because our task formulation is similar to the pre-train task of BART, where the entity-type special tokens can be seen as [MASK] tokens. Retaining the pre-training parameter is helpful to take the best advantage of PLM's knowledge.

Finally, there is a slightly negative effect when we reverse the parameter regions to save overlap and specific knowledge. We conjecture that $\text{APE}_{reversed}$ cannot model the order of knowledge utilization in the SC-RD framework.

Table 4: The performance of different learning tasks

Task	ACE05	RAMS	Wiki.
Joint EAE Task	69.9	49.4	64.1
PER Task	72.9	51.6	66.0

Table 5: The performance of different prompt styles

prompt style		ACE05	RAMS	Wiki.
overlap	specific			
ST	ST	72.9	51.6	66.0
NL	NL	72.1	51.1	65.3
NL	ST	69.5	49.3	63.5

4.3.2 Overlap Knowledge Learning Task

To investigate the effect of the PER task and its transparent training objective (Equation 8) in learning the overlap knowledge, we throw out the SC-RD framework and replace the PER task with Joint EAE Task like the previous work. The Joint EAE Task ignores the difference of datasets and merges multi-datasets to force the model directly learn overlap knowledge from the EAE training objective. The input and the target output string of the Joint EAE Task are as same as the specific knowledge learning phase. Two versions of Prefix will be respectively learned from the Joint EAE Task and the PER task and used in target datasets.

It can be observed in Table 4 that there is a 3.0%, 2.2%, and 1.9% decrease for the Arg-C F1 score on three datasets when changing the task. It is difficult for the model to discern the overlap knowledge from the imprecise EAE training objective. The PER task provides a transparent training objective to indicate the overlap knowledge explicitly.

4.3.3 Stressing Entity-Type Prompt

As aforementioned, prompts that keeping the same style in two learning phases can ignite the utilization of overlap knowledge in the specific knowledge learning phase and EAE inference scene. In

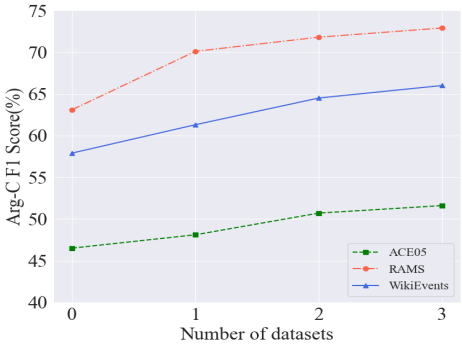

Figure 3: The performance of different multi-datasets

order to verify it, we propose another prompt style named **N**atural **L**anguage Pronouns (NL), which replaces the entity-type **S**pecial **T**oken (ST) with pronouns. The conversion between the two styles is shown in Appendix Table 9. We observe in Table 5 that there is a huge F1 score decrease of about 3.4% on the ACE05 dataset when we build prompts with different styles in two learning phases. The result indicates that narrowing the gap between the two phases is crucial to ignite the overlap knowledge. Meanwhile, the special token is a more powerful way to alert the model to the entity type than natural language.

4.3.4 Number of Datasets in Multi-datasets

In order to deeply observe the impact of the amount of the training data used in the overlap knowledge learning phase, we trained four versions of Prefix on varying numbers of training sets and transferred them to the target dataset. When the number of datasets was set to 0, the Prefix was randomly initialized and used directly without training. When the number of datasets was set to 1, we trained Prefix on {ACE05}. When the number of datasets was 2, we trained Prefix on {ACE05, RAMS}. Figure 3 shows the Arg-C F1 score increase as the number of datasets used to learn the overlap knowledge. The experiment result shows that with more available out-domain event records, the APE model can learn more abundant overlap knowledge and achieve better performance in the target dataset.

5 Related Works

5.1 Transfer Learning in EAE

Event argument extraction (EAE) aims to extract event arguments by the given event trigger and argument roles (Chen et al., 2015). Most existing approaches (Lin et al., 2020; Du and Cardie, 2020; Lu et al., 2021; Nguyen et al., 2022; Ma et al., 2022) suffer from insufficient training data and cannot perform better. Therefore, some studies (Liu et al., 2020b; Chen et al., 2020; Feng et al., 2020) focus on transferring knowledge from machine reading comprehension (MRC) datasets. Huang et al. (2022) leverages multilingual pre-trained models (Liu et al., 2020a; Xue et al., 2021) to achieve cross-lingual knowledge transfer. About transferring overlap knowledge from other available event datasets to the target dataset, only Zhou et al. (2022) attempt to introduce variational information bottleneck (Li and Eisner, 2019) to explore the overlap knowledge from two event datasets. Unlike their work, we clearly define the cross-dataset overlap knowledge in the EAE task. Our model does not limit the number of datasets and can explore overlap knowledge from all available datasets to achieve better performance.

5.2 Parameter-Efficient Tuning Method

Optimizing all the parameters of the PLMs means we need to save a complete fine-tuned model for every downstream task. The storage cost is prohibitively expensive with the increasing size of PLMs. Several parameter-efficient tuning methods (Houlsby et al., 2019; Hu et al., 2022; Mao et al., 2022; He et al., 2022) were proposed to mitigate this issue, which update a small number of task-specific parameters while keeping other pre-trained parameters frozen. Houlsby et al. (2019)equip each Transformer layer with an Adapter, and only the Adapters are tunable to save task-specific knowledge of the downstream task. Inspired by significant effectiveness achieved in prompt learning (Brown et al., 2020; Gao et al., 2021), Li and Liang (2021) prepends Prefix vectors to the hidden state, and only the Prefix will be trained on downstream tasks. To the best of our knowledge, we are the first to assemble two parameter-efficient tuning methods to separate knowledge in transfer learning and overcome catastrophic forgetting.

6 Conclusion

In this work, we first define the shareable overlap knowledge across datasets and reformulate the EAE task into two learning phases. Then, we propose APE model, which assembles two parameter-efficient tuning methods to save the overlap and specific knowledge. The experiment results show the efficiency of the cross-dataset transfer learning,

and APE achieves new SOTA with a large margin in the EAE task. Our model significantly reduces the need for new event records and achieves superior performance with few samples of target datasets. The ablation studies verify that our approach can explore overlap knowledge from multi-datasets and overcome the well-known catastrophic forgetting issue. In the future, we would like to study overlap knowledge across datasets in other information extraction tasks.

Limitations

This work introduces a pseudo-entity recognition (PER) task to supervise the model learning overlap knowledge. Since no additional entity annotation is available, we manually create a mapping function $\mathcal{M}(r)$, which maps each argument role r to an entity type. With the help of the mapping function $\mathcal{M}(r)$, the EAE label can be converted to the PER label. However, because the annotation of the EAE task is complicated, it is hard to avoid a few exceptional samples in the prior mapping function. Some entity words may be attached to impertinent entity types. For example, there is a triple of argument role, event type, and argument in RAMS's *movement.transportartifact.preventexit* event: **{Artifact, Object, Two pilots}**. The "Artifact" argument is mapped to "Object" in $\mathcal{M}(r)$, but we expect "Two pilots" can be mapped to "Person Or Organization". We tolerate such exceptional samples, and the occasional noise has not affected the training of APE.

Ethics Statement

Event argument extraction (EAE) task is a well-defined and classic task in Information Extract (IE) field. In this work, our use of existing artifacts (e.g., datasets) was licensed and consistent with their intended use. We do not see other significant ethical concerns. Our model is excepted to be used in extracting structured event records from plain text.

Acknowledgements

This work was supported by Beijing Natural Science Foundation(Grant No.4222032) and the Foundation for Innovative Research Groups of the National Natural Science Foundation of China(Grant No.61921003)

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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.

Yunmo Chen, Tongfei Chen, Seth Ebner, Aaron Steven White, and Benjamin Van Durme. 2020. Reading the manual: Event extraction as definition comprehension. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, pages 74–83, Online. Association for Computational Linguistics.

George R. Doddington, Alexis Mitchell, Mark A. Przybocki, Lance A. Ramshaw, Stephanie Strassel, and Ralph M. Weischedel. 2004. The automatic content extraction (ace) program - tasks, data, and evaluation. In *International Conference on Language Resources and Evaluation*.

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.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online. Association for Computational Linguistics.

Rui Feng, Jie Yuan, and Chao Zhang. 2020. Probing and fine-tuning reading comprehension models for few-shot event extraction. *CoRR*, abs/2010.11325.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. 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 3816–3830, Online. Association for Computational Linguistics.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Kuan-Hao Huang, I-Hung Hsu, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. Multilingual generative language models for zero-shot cross-lingual event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4633–4646, Dublin, Ireland. Association for Computational Linguistics.

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5405–5411, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 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.

Xiang Lisa Li and Jason Eisner. 2019. Specializing word embeddings (for parsing) by information bottleneck. 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 2744–2754, Hong Kong, China. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. 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 4582–4597, Online. 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.

Chunxi Liu, Qiaochu Zhang, Xiaohui Zhang, Kritika Singh, Yatharth Saraf, and Geoffrey Zweig. 2020a. Multilingual graphemic hybrid ASR with massive data augmentation. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 46–52, Marseille, France. European Language Resources association.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020b. 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.

Jian Liu, Yufeng Chen, and Jinan Xu. 2022a. Saliency as evidence: Event detection with trigger saliency attribution. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4573–4585, 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. 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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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.

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.

Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Scott Yih, and Madian Khabsa. 2022. UniPELT: A unified framework for parameter-efficient language model tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6253–6264, Dublin, Ireland. Association for Computational Linguistics.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press.

Minh Van Nguyen, Bonan Min, Franck Dernoncourt, and Thien Nguyen. 2022. Joint extraction of entities, relations, and events via modeling inter-instance and inter-label dependencies. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4363–4374, Seattle, United States. Association for Computational Linguistics.

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

Jiawei Sheng, Rui Sun, Shu Guo, Shiyao Cui, Jiangxia Cao, Lihong Wang, Tingwen Liu, and Hongbo Xu. 2022. Cored: Incorporating type-level and instance-level correlations for fine-grained event detection. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 1122–1132, New York, NY, USA. Association for Computing Machinery.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Jie Zhou, Qi Zhang, Qin Chen, Liang He, and Xuanjing Huang. 2022. A multi-format transfer learning model for event argument extraction via variational information bottleneck. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 1990–2000. International Committee on Computational Linguistics.

A Detail of Evaluation Metric

We adopt two widely-used evaluation metrics:

1. **Arg**ument **I**dentification F1 score (Arg-I): when the predicted argument's offsets match any of the gold argument labels in this event, we consider the predicted argument is correct.

2. **Arg**ument **C**lassification F1 score (Arg-C): when the predicted argument's argument role also matches the gold argument label, we consider the predicted argument is correct.

For the WikiEvents dataset, following Li et al. (2021), we add argument head F1 score (Head-C), which only focuses matching the headword of the arguments' offsets.

Table 6: The finite entity type set

Entity Type	Description	Example
Person Or Organization	The word that refers to a person or an organization	he, she, Bill, the president, ...
Location	The word that refers to a place or a region	Washinton DC, London, ...
Time	The word that indicates a time	10 June, 17 pm., ...
Money	The word that indicates money	$1,000, 6 million dollars, ...
Object	The word that refers to a materiality entity	The truck, bomb, gun, house, ...
Definition	The proper noun or immateriality entity	murder, crime of pillage

Table 7: The statistics of datasets, #Sent. is the number of sentences of the dataset, #Arg. is the number of arguments of the dataset.

Dataset	Train		Dev		Test	
	#Sent.	#Arg.	#Sent.	#Arg.	#Sent.	#Arg.
ACE05	17172	4859	923	605	832	576
RAMS	7329	17026	924	2188	871	2023
WikiEvents	5262	4552	378	428	492	566

Table 8: The performance of different variants on three datasets

Variant	Param		ACE05	RAMS	Wiki.
	overlap	specific			
APE	Prefix	Adapter	72.9	51.6	66.0
$\text{APE}_{reversed}$	Adapter	Perfix	72.1	51.2	64.7
w/o Prefix	BART	Adapter	71.5	51.3	64.3
w/o Adapter	Prefix	BART	71.7	50.9	64.8
BART	BART	BART	69.4	49.1	63.7

Table 9: The conversion between entity-type special token and natural language pronouns

Entity Type	Special Token	Natural Language Pronouns
Person Or Organization	[person/organization]	someone
Location	[location]	someplace
Time	[time]	some time
Money	[money]	some money
Object	[object]	something
Definition	[definition]	some definition

Table 10: Some examples of $\mathcal{M}(r)$ in three datasets, the complete $\mathcal{M}(r)$ can be found in our codebase.

Dataset	Event Type	Event Argument Role	Entity Type
ACE05	Business.Declare-Bankruptcy	Org	person/organization
		Place	location
		Time	time
	Business.End-Org	Place	location
		Org	person/organization
		Time	time
	Justice.Arrest-Jail	Person	person/organization
		Agent	person/organization
		Crime	definition
		Place	location
		Time	time
RAMS	transaction.transfermoney. purchase	recipient	person/organization
		beneficiary	person/organization
		money	money
		place	location
		giver	person/organization
	contact.mediastatement. broadcast	recipient	person/organization
		communicator	person/organization
		place	location
	movement.transportartifact. disperseseparate	artifact	object
		vehicle	object
		origin	location
		destination	location
		transporter	person/organization
WikiEvents	Contact.RequestCommand.Meet	Communicator	person/organization
		Recipient	person/organization
		Topic	definition
		Place	location
	Justice.ChargeIndict. Unspecified	Prosecutor	person/organization
		Defendant	person/organization
		JudgeCourt	person/organization
		Crime	definition
		Place	location
	Life.Die.Unspecified	Victim	person/organization
		Place	location
		Killer	person/organization
		MedicalIssue	definition

Table 11: Some examples of prompt in three datasets, the complete prompts can be found in our codebase.

Dataset	Event Type	Prompt
ACE05	Life.Die	[person/organization] killed [person/organization] with [object] at [location]
	Life.Injure	[person/organization] injured [person/organization] with [object] at [location]
	Justice.Fine	[person/organization] courted or judged fined [person/organization] at [location] for [definition] cost [money]
RAMS	conflict.attack.stabbing	[person/organization] attacked [person/organization] using [object] at [location]
	artifactexistence.damagedestroy.n/a	[person/organization] damaged or destroyed [object] using [object] in [location]
	movement.transportartifact.n/a	[person/organization] transported [object] in [object] from [location] place to [location] place
WikiEvents	Contact.Contact.Unspecified	[person/organization] communicated with [person/organization] about [definition] at [location]
	ArtifactExistence. ManufactureAssemble. Unspecified	[person/organization] manufactured or assembled or produced [object] from [object] using [object] at [location]
	Life.Illness.Unspecified	[person/organization] has [definition] sickness or illness at [location]

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
7: Limitations

☑ A2. Did you discuss any potential risks of your work?
6: Conclusion 7: Limitations

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract 1: Introduction

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

2: Method 3: Experiments Setup 4: Results and Analyses

☑ B1. Did you cite the creators of artifacts you used?
1: Introduction 2: Method 3: Experiments Setup 4: Results and Analyses 5: Related Works

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
3: Experiments Setup 8: Ethics Statement

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
3: Experiments Setup 8: Ethics Statement

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We didn't collect the information ourselves. The datasets we used are all widely used public datasets. Their content is mostly from news and we do not see any anonymization risk.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
3: Experiments Setup

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
3: Experiments Setup

C ☑ Did you run computational experiments?

3: Experiments Setup 4: Results and Analyses

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
3: Experiments Setup 4: Results and Analyses

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
3: Experiments Setup

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
3: Experiments Setup 4: Results and Analyses

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
3: Experiments Setup

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.