Separation and Fusion: A Novel Multiple Token Linking Model for Event Argument Extraction

Jing Xu[1], Dandan Song[1*], Siu Cheung Hui[2], Zhijing Wu[1], Meihuizi Jia[1], Hao Wang[1], Yanru Zhou[1], Changzhi Zhou[1], Ziyi Yang[3]

[1]School of Computer Science and Technology, Beijing Institute of Technology, China
[2]Nanyang Technological University, Singapore
[3]School of Cyberspace Science and Technology, Beijing Institute of Technology, China
{xujing,sdd,zhijingwu,jmhuizi24,wanghaobit,zhouyanru,
zhou_changzhi97}@bit.edu.cn;asschui@ntu.edu.sg;yziyi@bit.edu.cn

Abstract

In event argument extraction (EAE), a promising approach involves jointly encoding text and argument roles, and performing multiple token linking operations. This approach further falls into two categories. One extracts arguments within a single event, while the other attempts to extract arguments from multiple events simultaneously. However, the former lacks to leverage cross-event information and the latter requires tougher predictions with longer encoded role sequences and extra linking operations. In this paper, we design a novel separation-and-fusion paradigm to separately acquire cross-event information and fuse it into the argument extraction of a target event. Following the paradigm, we propose a novel multiple token linking model named Sep2F, which can effectively build event correlations via roles and preserve the simple linking predictions of single-event extraction. In particular, we employ one linking module to extract arguments for the target event and another to aggregate the role information of multiple events. More importantly, we propose a novel two-fold fusion module to ensure that the aggregated cross-event information serves EAE well. We evaluate our proposed model on sentence-level and document-level datasets, including ACE05, RAMS, WikiEvents and MLEE. The extensive experimental results indicate that our model outperforms the state-of-the-art EAE models on all the datasets.

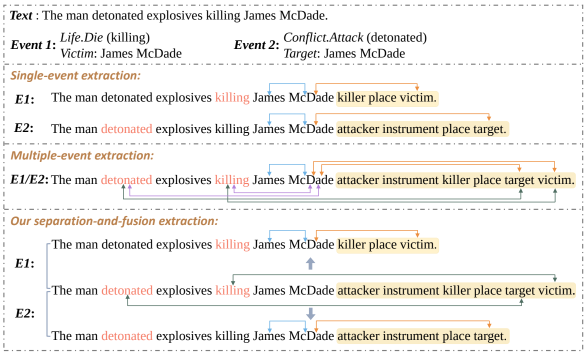

Figure 1: Different categories of multiple token linking models for EAE. *E1*: *Event 1*. *E2*: *Event 2*. The trigger words are highlighted in red, the concatenated roles are in yellow, and the arrows of different colors represent different linking operations. Note that we only exhibit one argument of each event for simplification.

1 Introduction

As a crucial step of event extraction (EE), event argument extraction (EAE) aims to recognize all arguments and their roles for each event in text. The recognized arguments can act as structured semantic information and greatly influence various downstream tasks (Wen et al., 2021; Wu et al., 2022; Fung et al., 2023; Liu et al., 2023b). Despite the impressive advancements in large language models (LLMs) such as ChatGPT (OpenAI, 2022), the evaluations (Han et al., 2023; Li et al., 2023a; Wei et al., 2023) indicate that EAE remains challenging.

Recently, significant improvements have been made in EAE using prompt-based methods by extractive (Ma et al., 2022; He et al., 2023; Nguyen et al., 2023; Li et al., 2023b) and generative (Hsu et al., 2022; Du et al., 2022; Zhang et al., 2023) styles. However, the former is limited by predetermined numbers of repeated role slots when extracting multiple arguments with the same role, while the latter is weak in accommodating long-distance argument extraction. Besides, most of their performance relies on the quality of their designed prompts.

Different from the prompt-based methods, several works (Wang et al., 2022; Lou et al., 2023; Liu et al., 2023a) concatenate input text and argument roles as a natural language sequence, jointly encode them, and perform multiple token linking operations for EAE or universal information extraction. The direct and parallel linking operations between

*Corresponding author.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6611–6624

all arguments and roles make extracting multiple arguments with the same role straightforward, and promote the interaction among scattered arguments within the long text. Additionally, they do not need well-designed prompts. Based on the number of events extracted at a time, we can further divide these methods into two categories: *single-event extraction* and *multiple-event extraction*.

The first category of methods (Wang et al., 2022; Liu et al., 2023a) involves concatenating text and event-specific roles as input, followed by two linking operations for each event. As shown in Figure 1, the argument "*James McDade*" plays different roles in both the "*Life.Die*" and "*Conflict.Attack*" events and is extracted separately. Though *single-event extraction* facilitates simple linking predictions, it ignores the significant correlations across different events (Zeng et al., 2022; He et al., 2023; Li et al., 2023b). The second category (Lou et al., 2023) attempts to extract arguments of multiple events simultaneously. Nevertheless, two issues come with this *multiple-event extraction*. First, the roles concatenated with the input text are no longer of a single event but instead of all involved events, making it more challenging to choose correct roles for arguments via link predictions. Second, compared with *single-event extraction*, two extra linking operations are needed to determine which event (trigger) the extracted arguments and corresponding roles belong to. As a result, they may accumulate more errors during the prediction process. Figure 1 demonstrates that to extract the argument "*James McDade*" from both the "*Life.Die*" and "*Conflict.Attack*" events simultaneously, the extra trigger-argument and trigger-role linking operations are essential.

To leverage both merits of the above two categories, we design a novel separation-and-fusion paradigm by (1) separating the cross-event information acquisition and the EAE process and (2) fusing the acquired cross-event information into EAE. Therefore, the final EAE can simultaneously preserve the simple linking predictions of *single-event extraction* and leverage the cross-event clues like *multiple-event extraction*. Figure 1 illustrates the paradigm. The middle part is separated from the EAE process and acquires cross-event information via trigger-role linkings. The upward and downward arrows denote the cross-event information fusion.

Following the paradigm, we propose a novel multiple token linking model with **Sep**arate acquisition of cross-event information and **Two**-fold **F**usion for EAE, named **Sep2F**. To separate the cross-event information acquisition and the argument extraction process, we design two multiple token linking modules. Specifically, we introduce one linking module to bridge each event trigger and its co-occurred roles for multiple events. Thus, the representations of different event triggers aggregate their co-occurred roles in parallel and provide critical cross-event information. Simultaneously, we employ another linking module to extract arguments for a target event. It performs two linking operations to obtain the argument spans and corresponding roles. More importantly, we propose a novel two-fold fusion module to effectively fuse the acquired cross-event information into the argument extraction for the target event. In details, we first dynamically fuse the text representations from the above two linking modules. Then, we utilize the fused text representations to obtain cross-module token linking scores. The linking scores are further fused into the final prediction scores. These two sequential fusions affect each other and deliver significant performance contributions. We summarize our main contributions as follows:

- We propose a novel separation-and-fusion paradigm for EAE. It can leverage cross-event information and retain the merits of single-event extraction simultaneously.
- Under the separation-and-fusion paradigm, we propose Sep2F, a novel multiple token linking model. Specifically, we design two linking modules to acquire cross-event information and extract arguments of a target event. Also, we introduce a two-fold fusion module to ensure that the acquired cross-event information serves the argument extraction well.
- We conduct extensive experiments on the widely used benchmarks, including ACE05, RAMS, WikiEvents and MLEE. Our proposed model outperforms the state-of-the-art EAE models by 2.0%, 1.0%, 2.8% and 2.5% in Arg-C F1, respectively.

2 Related Work

2.1 Event Argument Extraction

Earlier EAE methods mainly fall into two classes: classification-based methods and MRC-based methods. The former treats entity mentions (Wang et al., 2019; Ma et al., 2020; Xiangyu et al., 2021) or identified text spans (Ma et al., 2020; Ebner et al., 2020; Xu et al., 2022b) as argument candidates and employs randomly initialized classifiers to recognize argument roles. The latter designs question templates for argument roles and considers EAE as a machine reading comprehension problem (Du and Cardie, 2020; Liu et al., 2020; Li et al., 2020; Wei et al., 2021). Lately, prompt-based methods have delivered impressive performance improvements. Specifically, the extractive ones locate role slots in prompts to mine prior knowledge from pre-trained language models for EAE (Lin and Chen, 2021; Zhang et al., 2022; Ma et al., 2022; He et al., 2023; Li et al., 2023b). As for the generative ones, they leverage prompt templates and transformer-based encoder-decoder frameworks to extract the arguments within each event sequentially (Li et al., 2021; Hsu et al., 2022; Du et al., 2022; Zeng et al., 2022; Hsu et al., 2023; Zhang et al., 2023).

Most of the above methods ignore the correlations across different events and only a few works (Zeng et al., 2022; Du et al., 2022; He et al., 2023; Li et al., 2023b) consider the benefits from them. However, these methods are limited by the quality of prompts or degraded performance in long-range extraction. Therefore, this paper proposes a novel multiple token linking model that can capture cross-event information well and avoid these limitations.

2.2 Multiple Token Linking

Recently, a rising interest has emerged in multiple token linking models for information extraction, which jointly encode text and task-specific labels as a natural language sequence and perform token linking operations. UniRel (Tang et al., 2022) proposes entity-entity and entity-relation linking operations to extract relational triples. Wang et al. (2022) design a multiway attention mechanism to connect roles with argument candidates within a single event for EAE. RexUIE (Liu et al., 2023a) uses different token linking operations to identify each event's argument spans and role types separately in the universal information extraction framework. Unlike RexUIE, USM (Lou et al., 2023) employs extra trigger-argument and trigger-role linkings to determine which event the extracted arguments belong to for multiple-event extraction.

3 Proposed Model

We represent an instance as (X, T, C), where X is the input text, T denotes the target event and C denotes the other events surrounding T within the text X. Specifically, T is further represented as $(e, t, \mathcal{R}^e)$, where e is the event type, t is the trigger word and $\mathcal{R}^e$ is the set of role types specific to e. Similarly, C is represented as $\{(\tilde{e}_i, \tilde{t}_i, \mathcal{R}^{\tilde{e}_i})|\ i \leq |C|\}$ which provides the cross-event information. EAE aims to extract the argument set $\mathcal{A}$ for the target event T, where each argument $a^{(r)}$ in $\mathcal{A}$ is a snippet of the text X with the role type $r \in \mathcal{R}^e$.

As illustrated in Figure 2, we introduce our proposed model Sep2F, which consists of three modules: Linking Construction for Multiple Events, Linking Construction for Target Event and Two-fold Fusion. Sep2F follows our designed separation-and-fusion paradigm. Specifically, the first two modules separate the acquisition of cross-event information and the argument extraction process of the target event. In contrast, the last module introduces our proposed fusion for the acquired cross-event information. Next, we describe these modules in details.

3.1 Linking Construction for Multiple Events

In this module, we acquire cross-event information by aggregating the roles of multiple events, including the target event T and all surrounding events C. To achieve the goal, we build the connections between different event triggers and their corresponding involved roles in parallel. First, we jointly encode text and all roles pre-defined in the given dataset. Then, we introduce the label matrix and score matrix for multiple trigger-role linkings within these events. Finally, we formulate the training loss in this module.

Encoding We first verbalize each argument role as its role name, i.e., a single natural description word. For the few roles whose names contain multiple words, we employ additional special tokens to represent them. Note that the special token representations are not event-specific. Then, we concatenate all the verbalized roles pre-defined in the dataset as R_M. After that, we jointly encode the sequence R_M and the input text X with a pre-trained language model (PLM) as follows:

$$\mathbf{E}^M = (\mathbf{h}_1^M, \cdots, \mathbf{h}_N^M, \mathbf{r}_1^M, \cdots, \mathbf{r}_{|R_M|}^M) = \text{PLM}(X \oplus R_M) \quad (1)$$

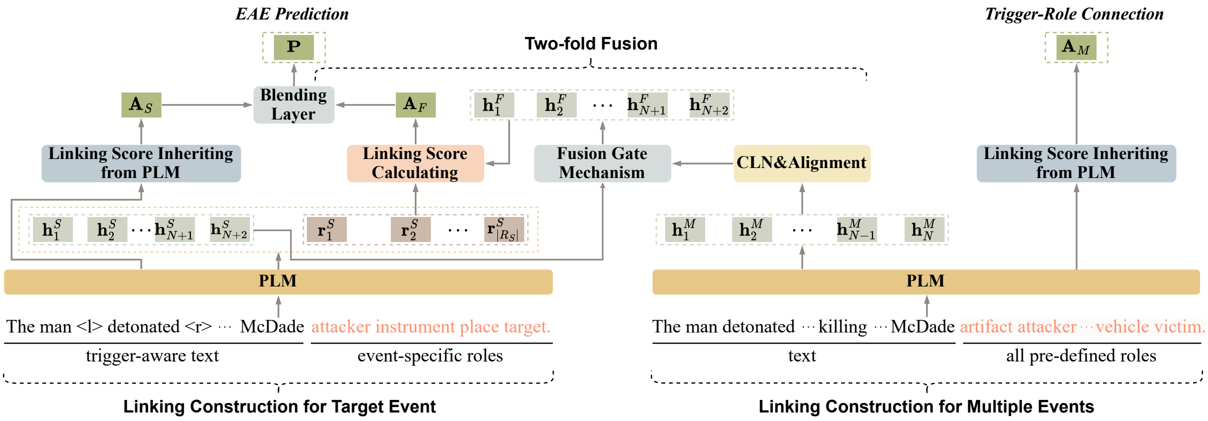

Figure 2: The overall architecture of our proposed model.

where $\mathbf{h}_n^M$ $(1 \le n \le N)$ and $\mathbf{r}_n^M$ $(1 \le n \le |R_M|)$ are the token embeddings within X and R_M, respectively.

Label Matrix To learn the respective role information for the multiple events, we design a label matrix $\mathbf{L}_M \in \mathbb{B}^{(N+|R_M|)\times(N+|R_M|)}$. For each event in T or C, we assume that the start and end token embeddings of its trigger word are $\mathbf{h}_i^M$ and $\mathbf{h}_j^M$, respectively. Further, for each role involved in the event, we assume that the token embedding is $\mathbf{r}_k^M$. Then, we conduct linking operations to bridge the trigger and the involved role. In details, we construct the linking pairs $(i, k+N)$ and $(k+N, j)$, and set $L_M[i][k+N]$ and $L_M[k+N][j]$ as *True*. For those token pairs that are not linked, we mark the corresponding values in $\mathbf{L}_M$ as *False*.

Score Matrix Following Tang et al. (2022), we inherit the multi-head self-attention results of the transformer-based PLM as the linking scores between token pairs. Specifically, we average the multi-head self-attention weights without Softmax-normalization from the last layer of the PLM:

$$\mathbf{A}_M = \frac{1}{P}\sum_{p}^{P}\frac{\mathbf{Q}_p\mathbf{K}_p^\top}{\sqrt{d_h}} \tag{2}$$

where P is the number of heads, d_h is the dimension of queries and keys, and $\mathbf{Q}_p$ and $\mathbf{K}_p$ are the query and key matrices, respectively. $\mathbf{A}_M \in \mathbb{R}^{(N+|R_M|)\times(N+|R_M|)}$ represents the trigger-role linking scores for the multiple events.

Training Loss We use the loss $\mathcal{L}_{\text{TR}}$ to learn the auxiliary task, which guides the connections between different event triggers and their corresponding involved roles as follows:

$$\mathcal{L}_{\text{TR}} = -\frac{1}{(N+|R_M|)^2}\sum_i\sum_j\Big(L_{i,j}^M \log\sigma\left(A_M[i][j]\right) + (1-L_{i,j}^M)\log\left(1-\sigma\left(A_M[i][j]\right)\right)\Big) \tag{3}$$

where σ denotes a sigmoid function. $L_{i,j}^M$ is set to 1 when $L_M[i][j]$ is *True*, while $L_{i,j}^M$ is set to 0 if otherwise.

3.2 Linking Construction for Target Event

To extract arguments for the given target event $(e, t, \mathcal{R}^e)$, we jointly encode text and event-specific roles, define the label matrix and score matrix for multiple token linkings and present the training loss in this module.

Encoding We verbalize and concatenate all argument roles in the role set $\mathcal{R}^e$ as a token sequence R_S. Then, we follow Ma et al. (2022) to insert two special tokens $\langle \ell \rangle$ and $\langle r \rangle$ into the text X to mark the position of the trigger t:

$$X_S = (x_1, \cdots, \langle \ell \rangle, t, \langle r \rangle, \cdots, x_{|X|}) \tag{4}$$

After that, we concatenate the trigger-aware text X_S and the sequence R_S and leverage another PLM to encode them:

$$\mathbf{E}^S = (\mathbf{h}_1^S, \cdots, \mathbf{h}_{N+2}^S, \mathbf{r}_1^S, \cdots, \mathbf{r}_{|R_S|}^S) = \text{PLM}(X_S \oplus R_S) \tag{5}$$

where $\mathbf{h}_n^S$ $(1 \le n \le N+2)$ and $\mathbf{r}_n^S$ $(1 \le n \le |R_S|)$ are the token embeddings within X_S and R_S, respectively.

Label Matrix We define a label matrix to tag argument spans and roles in the given target event, denoted as $\mathbf{L}_S \in \mathbb{B}^{(N+|R_S|+2)\times(N+|R_S|+2)}$. For each argument, we assume that its role embedding is $\mathbf{r}_k^S$, and its start and end token embeddings are $\mathbf{h}_i^S$ and $\mathbf{h}_j^S$, respectively. We first construct the linking pairs (i, j) and (j, i) to tag the argument span, and set $L_S[i][j]$ and $L_S[j][i]$ as *True*. Meanwhile, we tag the role information by argument-role linking operations. In details, we employ two linking pairs $(i, k+N+2)$ and $(k+N+2, j)$, and set $L_S[i][k+N+2]$ and $L_S[k+N+2][j]$ as *True*. Besides, we mark the values in $\mathbf{L}_S$, whose corresponding token pairs are not linked, as *False*.

Score Matrix Following Equation (2), we also employ the averaged multi-head self-attention weights from the last layer of the PLM used in the module as the token linking scores, denoted as $\mathbf{A}_S \in \mathbb{R}^{(N+|R_S|+2)\times(N+|R_S|+2)}$.

Training Loss To leverage the acquired cross-event information, we first employ the two-fold fusion, which will be described in the next module in details, to obtain the final token linking prediction matrix:

$$\mathbf{P} = \text{TFF}(\mathbf{E}^M, \mathbf{E}^S, \mathbf{A}_S) \tag{6}$$

where TFF refers to the two-fold fusion. Then, we obtain the loss $\mathcal{L}_{\text{EAE}}$ as follows:

$$\begin{aligned}\mathcal{L}_{\text{EAE}} = -\frac{1}{(N+|R_S|+2)^2}\sum_i\sum_j \Big(L_{i,j}^S \log P[i][j] + \\ (1-L_{i,j}^S)\log(1-P[i][j])\Big)\end{aligned} \tag{7}$$

where $L_{i,j}^S$ is set to 1 when $L_S[i][j]$ is *True*, while $L_{i,j}^S$ is set to 0 if otherwise.

3.3 Two-fold Fusion

In this module, we utilize the learned text representations with aggregated cross-event role information to enhance EAE for the target event. Specifically, we first dynamically fuse the text representations from the above two linking modules, referred to as the first-fold fusion. Then, we use the fused text representations to obtain another token linking score matrix for the target event, different from $\mathbf{A}_S$. These two token linking score matrices are further fused as the final linking predictions, referred to as the second-fold fusion.

First-fold Fusion As the encoded text embeddings in $\mathbf{E}^M$ are involved with multiple events, we first leverage conditional layer normalization (CLN) (Su, 2019; Yu et al., 2021; Xu et al., 2022a) to generate the contextual embeddings for the target event (trigger). For the target trigger t, we simply feed it into a frozen PLM and employ the first token embedding to represent it, denoted as $\mathbf{t}$. For each token embedding $\mathbf{h}_n^M$ $(1 \leq n \leq N)$ in $\mathbf{E}^M$, the integrated embedding $\hat{\mathbf{h}}_n^M$ $(1 \leq n \leq N)$ is acquired as follows:

$$\alpha_t = \mathbf{t}\mathbf{W}_\alpha + \mathbf{b}_\alpha \tag{8}$$

$$\beta_t = \mathbf{t}\mathbf{W}_\beta + \mathbf{b}_\beta \tag{9}$$

$$\hat{\mathbf{h}}_n^M = \text{CLN}(\mathbf{h}_n^M, \alpha_t, \beta_t) = \alpha_t \odot \left(\frac{\mathbf{h}_n^M - \mu}{\sigma}\right) + \beta_t \tag{10}$$

where $\mathbf{W}_\alpha \in \mathbb{R}^{d_1\times d_1}$, $\mathbf{W}_\beta \in \mathbb{R}^{d_1\times d_1}$, $\mathbf{b}_\alpha \in \mathbb{R}^{d_1}$ and $\mathbf{b}_\beta \in \mathbb{R}^{d_1}$ are trainable parameters, μ and σ are the mean and standard deviation calculated across the elements of $\mathbf{h}_n^M$, respectively. Note that d_1 is the dimension of $\mathbf{t}$. Then, $(\hat{\mathbf{h}}_1^M, \cdots, \hat{\mathbf{h}}_N^M)$ are inserted with the zero vector $\mathbf{0}$ to facilitate the alignment with the text embeddings in $\mathbf{E}^S$:

$$(\bar{\mathbf{h}}_1^M, \cdots, \bar{\mathbf{h}}_{N+2}^M) = (\hat{\mathbf{h}}_1^M, \cdots, \mathbf{0}, \hat{\mathbf{h}}_i^M, \cdots, \hat{\mathbf{h}}_j^M, \mathbf{0}, \cdots, \hat{\mathbf{h}}_N^M) \tag{11}$$

where i and j correspond to the start and end positions for the token embeddings of the trigger t in $(\mathbf{h}_1^S, \cdots, \mathbf{h}_{N+2}^S)$. Finally, we dynamically fuse the different text embeddings. Given two token embeddings $\mathbf{h}_n^S$ and $\bar{\mathbf{h}}_n^M$ $(1 \leq n \leq N+2)$, we use a fusion gate mechanism to obtain the token embedding $\mathbf{h}_n^F$ as follows:

$$\mathbf{g}_n = \sigma\left(\left[\mathbf{h}_n^S; \bar{\mathbf{h}}_n^M\right]\mathbf{W}_G\right) \tag{12}$$

$$\mathbf{h}_n^F = \mathbf{g}_n \odot \mathbf{h}_n^S + (1-\mathbf{g}_n) \odot \bar{\mathbf{h}}_n^M \tag{13}$$

where $\odot$ denotes the element-wise multiplication operation and $\mathbf{W}_G \in \mathbb{R}^{2d_1\times d_1}$ is a trainable matrix. Note that d_1 is the dimension of $\mathbf{h}_n^S$ and $\bar{\mathbf{h}}_n^M$.

Second-fold Fusion We concatenate the fused text embeddings $(\mathbf{h}_1^F, \cdots, \mathbf{h}_{N+2}^F)$ and the role embeddings $(\mathbf{r}_1^S, \cdots, \mathbf{r}_{|R_S|}^S)$, denoted as $\mathbf{F}$. Then, we calculate another token linking score matrix $\mathbf{A}_F \in \mathbb{R}^{(N+|R_S|+2)\times(N+|R_S|+2)}$ as follows:

$$\mathbf{F}_Q = \mathbf{F}\mathbf{W}_Q,\ \mathbf{F}_K = \mathbf{F}\mathbf{W}_K,\ \mathbf{A}_F = \mathbf{F}_Q\mathbf{F}_K^\top \tag{14}$$

where $\mathbf{W}_Q \in \mathbb{R}^{d_1\times d_2}$ and $\mathbf{W}_K \in \mathbb{R}^{d_1\times d_2}$ are trainable matrices. After that, we fuse the two token linking score matrices $\mathbf{A}_S$ and $\mathbf{A}_F$ using a blending layer (Wolpert, 1992):

$$\mathbf{P} = \sigma(\mathbf{A}_S + \mathbf{A}_F - \tau) \tag{15}$$

where τ is a trainable parameter.

Algorithm 1 Inference Process

Input: Predicted set $\mathcal{C}((s_k, e_k)), k \in (1, \cdots, |\mathcal{C}|)$.
Output: Argument set $\mathcal{A}$.

```
Let cand_span = set().
Let start2r_dict = {}, end2r_dict = {}.
for (s_k, e_k) ∈ C do
    if s_k ≤ N + 2 and e_k ≤ N + 2 then
        cand_span.add((s_k, e_k))
    else if s_k ≤ N + 2 and e_k > N + 2 then
        start2r_dict[s_k].append(e_k − N − 2)
    else if s_k > N + 2 and e_k ≤ N + 2 then
        end2r_dict[e_k].append(s_k − N − 2)
    end if
end for
for (s, e) ∈ cand_span do
    for r ∈ (set(start2r_dict[s]) ∩ set(end2r_dict[e]))
    do
        A.add((s, e, r))
    end for
    for r ∈ (set(start2r_dict[e]) ∩ set(end2r_dict[s]))
    do
        A.add((e, s, r))
    end for
end for
```

3.4 Training and Inference

Training We introduce the overall training loss as follows:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{EAE}} + (1-\alpha)\mathcal{L}_{\text{TR}} \tag{16}$$

where α $(0 < \alpha < 1)$ represents a weight hyperparameter.

Inference Based on the token linking prediction matrix $\mathbf{P}$, we first add each linking pair (i, j) to the predicted linking pair set $\mathcal{C}$ when its prediction value $P[i][j](1 \leq i, j \leq N + |R_S| + 2)$ exceeds a threshold hyperparameter, denoted as δ. Then, we employ $\mathcal{C}$ as input to run the inference algorithm as shown in Algorithm 1 and obtain the extracted argument set $\mathcal{A}$ of the target event. For each item $(start, end, role)$ in $\mathcal{A}$, $start$ and end denote the start and end positions of an argument, respectively, and $role$ is the index of the argument role in R_S.

4 Performance Evaluation

4.1 Experimental Setup

Datasets To evaluate our proposed model, we conduct experiments on one sentence-level dataset ACE05 (Doddington et al., 2004) and three document-level datasets, including RAMS (Ebner et al., 2020), WikiEvents (Li et al., 2021) and MLEE (Pyysalo et al., 2012). Following Ma et al. (2022), we preprocess ACE05 by using the scripts of DyGIE++ (Wadden et al., 2019). As for the document-level datasets, we follow He et al. (2023) to employ a predefined window length, which is set to 250, to split each document into context segments. See the dataset details in Appendix A.

Metrics We follow previous works (Ma et al., 2022; He et al., 2023) to adopt two metrics to measure the performance. (1) Argument Identification F1 (Arg-I): Regard an argument of an event identified correctly when its boundary agrees with any golden arguments of the event. (2) Argument Classification F1 (Arg-C): Regard an argument of an event classified correctly when its boundary and role agree with any golden arguments of the event.

Implementation Details For a fair comparison with recent works, we leverage RoBERTa-base and RoBERTa-large (Liu et al., 2019) as the PLM in our model. Specifically, we train the base model on one NVIDIA RTX 3090 24G GPU and the large model on one NVIDIA A100 40G GPU. The Adam optimizer with a linear learning rate scheduler and the warmup strategy with a ratio of 0.1 are adopted. As δ serves as the inference threshold of the binary classification prediction value $P[i][j]$ in Equation (7), we follow the most binary classification setting to set δ as 0.5 for all the datasets. For the other hyperparameters, we attach the details in Appendix B.

Baselines We compare our Sep2F with the following models, all of which evaluate the EAE performance on **both sentence-level and document-level datasets**: EEQA (Du and Cardie, 2020), BART-Gen (Li et al., 2021), PAIE (Ma et al., 2022), EDGE (Li et al., 2023b), APE(Single) (Zhang et al., 2023), TabEAE (He et al., 2023). Moreover, ChatGPT equipped with in-context learning (ICL) (Brown et al., 2020) has presented impressive performance in various NLP tasks. Still, there needs to be a comprehensive evaluation of different EAE datasets, especially document-level datasets. Thus, we follow Han et al. (2023) to construct 5-shot ICL prompts as input for each test sample and use the OpenAI API access[1] to acquire the EAE results. Specifically, we evaluate the EAE performance with two versions of ChatGPT: *gpt-3.5-turbo* and *gpt-4*. See Appendix D for detailed prompt construction.

Additionally, we notice there are a few models, including UnifiedEAE (Zhou et al., 2022) and APE (Zhang et al., 2023), which focus on leveraging multiple datasets to enhance the performance

[1]https://platform.openai.com/

Model	PLM	ACE05		RAMS		WikiEvents		MLEE	
		Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
ChatGPT(5-shot ICL)	GPT-3.5	35.6	30.0	25.1	19.3	13.6	11.6	15.7	11.8
ChatGPT(5-shot ICL)	GPT-4	37.5	33.2	23.8	20.9	16.7	15.3	16.9	14.9
EEQA (Du and Cardie, 2020)*	RoBERTa-l	72.1	70.4	51.9	47.5	60.4	57.2	70.3	68.7
BART-Gen (Li et al., 2021)	BART-l	69.9	66.7	51.2	47.1	66.8	62.4	71.0	69.8
PAIE (Ma et al., 2022)	BART-l	75.7	72.7	56.8	52.2	70.5	65.3	72.1	70.8
PAIE (Ma et al., 2022)*	RoBERTa-l	76.1	73.0	57.1	52.3	70.9	65.5	72.5	71.4
EDGE (Li et al., 2023b)	BART-b	75.3	70.6	55.2	49.7	68.2	62.8	-	-
APE(Single) (Zhang et al., 2023)	BART-l	75.3	72.9	56.3	51.7	70.6	65.8	-	-
TabEAE (He et al., 2023)	RoBERTa-l	<u>77.2</u>	<u>75.0</u>	<u>57.3</u>	<u>52.7</u>	71.4	66.5	75.1	74.2
Sep2F (Ours)	RoBERTa-b	76.2	73.5	56.6	52.1	<u>73.3</u>	<u>68.2</u>	<u>76.8</u>	<u>75.6</u>
Sep2F (Ours)	RoBERTa-l	**78.8**	**77.0**	**58.7**	**53.7**	**74.0**	**69.3**	**77.5**	**76.7**

Table 1: Experimental results based on four datasets. The best score is in bold and the second best score is underlined. * indicates the results from He et al. (2023). b and l in the column PLM represent the base and large models, respectively. Note that we report the averaged results of our Sep2F with three different fixed random seeds.

for a target dataset and benefit from the extra resources. Thus, we exclude them from our main results but leave the comparison in Appendix E.

4.2 Results and Analysis

Main Results As shown in Table 1, we first summarize that our large model achieves new state-of-the-art (SOTA) performance on all the datasets. Specifically, compared with the latest SOTA model TabEAE, our large model obtains **1.6%/2.0%**, **1.4%/1.0%**, **2.6%/2.8%** and **2.4%/2.5%** absolute improvements in Arg-I/Arg-C F1 on ACE05, RAMS, WikiEvents and MLEE, respectively. Moreover, our base model averagely exceeds EDGE, which only leverages base PLMs, by 2.5%/3.6% in Arg-I/Arg-C F1 on the three used datasets. Also, using the smaller PLM, our base model outperforms all the baselines powered by large PLMs on WikiEvents and MLEE and obtains competitive performance on ACE05 and RAMS. The results indicate our proposed Sep2F exhibits outstanding performance in handling EAE of different levels.

In addition, we find that both versions of ChatGPT significantly fall behind existing supervised EAE models. The performance gap is usually extended when dealing with the more challenging document-level EAE task. Therefore, how to push LLMs such as ChatGPT to achieve comparable performance for EAE remains to be explored.

Comparison with Different Token Linking Models As mentioned in Introduction, there are mainly two existing categories of methods performing multiple token linking operations for EAE: *single-event extraction* and *multiple-event extraction*. However, these methods solve EAE as a subpart of end-to-end universal information extraction or event extraction tasks. Thus, their experimental settings and training datasets differ from handling EAE alone. For a fair comparison, we follow the details of these token linking methods to design the following variants: (1) **SingleE** trains the Linking Construction for Target Event module without the two-fold fusion to extract arguments of each single event. It corresponds to *single-event extraction*. (2) **MultiE** extracts arguments of multiple events simultaneously, which corresponds to *multiple-event extraction*. Specifically, we formulate such EAE as a multiple <trigger-role-argument> triple extraction[2] and refer to the implementation[3] of UniRel (Tang et al., 2022), which conducts their relational triple extraction with multi-token entities. Note that the roles concatenated with input text are only specific to the involved events. (3) **MultiE-L** concatenates all pre-defined roles of the

Model	ACE05 [1.35]	RAMS [1.25]	WikiEvents [1.78]	MLEE [3.32]
SingleE	75.4	53.0	66.5	74.4
MultiE	69.9	49.3	47.9	57.2
MultiE-L	68.1	40.5	26.5	8.4
Sep2F	**77.0**	**53.7**	**69.3**	**76.7**

Table 2: Performance comparison in Arg-C F1 (%) between different multiple token linking models. We present the average number of events per instance for each dataset in [·]. The performance is reported based on RoBERTa-large.

[2]As golden triggers are provided in EAE, we correct the span prediction errors of triggers during the inference process.

[3]https://github.com/wtangdev/UniRel

Model	PLM	ACE05		RAMS		WikiEvents		MLEE	
		$\|C\| = 0$ [185]	$\|C\| > 0$ [218]	$\|C\| = 0$ [587]	$\|C\| > 0$ [284]	$\|C\| = 0$ [114]	$\|C\| > 0$ [251]	$\|C\| = 0$ [175]	$\|C\| > 0$ [2025]
SingleE	RoBERTa-l	74.8	76.0	52.9	53.3	68.4	65.5	83.1	73.8
MultiE	RoBERTa-l	70.8	69.1	50.0	47.5	45.5	49.1	53.7	57.5
PAIE	RoBERTa-l	71.0	73.9	52.7	52.1	65.3	65.4	78.9	70.1
TabEAE	RoBERTa-l	73.4	76.1	52.9	52.5	67.3	66.2	81.1	73.6
Sep2F (Ours)	RoBERTa-b	72.8	74.0	52.1	52.0	66.8	69.0	**85.5**	74.9
Sep2F (Ours)	RoBERTa-l	**75.4**	**78.3**	**53.2**	**54.6**	**68.8**	**69.5**	84.3	**76.1**

Table 3: Arg-C F1 (%) comparison on test instances with different numbers of surrounding events. $|C|$ refers to the number of surrounding events for the target extracted event. The value in $[\cdot]$ denotes the corresponding number of test instances.

Model	ACE05	RAMS	WIKI	MLEE
Sep2F	77.0	53.7	69.3	76.7
- First-fold Fusion	74.8	51.2	66.3	75.6
- Second-fold Fusion	72.5	50.2	67.8	74.2

Table 4: Ablation results of the two-fold fusion. We report the performance in Arg-C F1 (%) and abbreviate WikiEvents as WIKI.

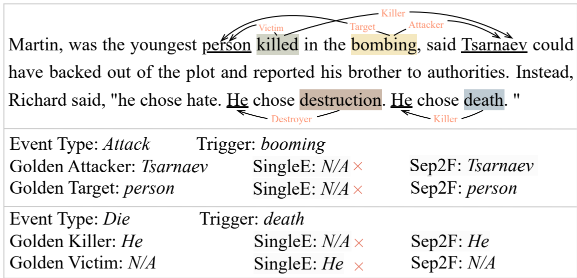

Figure 3: A case study from WikiEvents. The triggers of different events are in different colors. Note that the involved models are based on RoBERTa-large.

dataset with input text. As for the other settings, it follows MultiE.

From the results in Table 2, we conclude that our model surpasses all variants on the four datasets. Compared with our Sep2F, the performance of SingleE, which does not leverage cross-event information, drops by 1.6%, 0.7%, 2.8% and 2.3% in Arg-C F1 on ACE05, RAMS, WikiEvents and MLEE, respectively. Further, we see that MultiE fails to acquire competitive performance for EAE. Especially on WikiEvents and MLEE, the performance gap between MultiE and our model is rather significant. The main reason is that the average number of events per instance on these two datasets is larger than the others. As a result, MultiE struggles to handle more token linking predictions for each instance. Besides, we observe that MultiE-L generally lags behind MultiE. As expected, the more concatenated roles make the correct role choices more challenging when performing linking predictions between text and roles. Hence, a longer role concatenation will impair the performance of multiple token linking models.

Detailed Results on Single/Multiple Events Following He et al. (2023), we divide the test instances of each dataset into two groups according to the number of events (i.e., $|C|$) surrounding the target extracted event. When $|C| = 0$, only the target extracted event is in the instance. On the other hand, if $|C| > 0$, there are multiple events. Then, we investigate the detailed results based on the groupings. As illustrated in Table 3, we can observe that: (1) On both groups, the Arg-C F1 of our large model exceeds the previous SOTA model TabEAE. We attribute this improvement to our separation-and-fusion paradigm, which preserves both advantages of *single-event extraction* and *multiple-event extraction*. (2) Our base model demonstrates impressive performance results for the instances containing multiple events on WikiEvents and MLEE, even outperforming the two most competitive models utilizing large-version PLMs, SingleE and TabEAE. It verifies our model is good at handling instances with multiple events. (3) SingleE exhibits a comprehensive performance improvement compared with PAIE, despite both disregarding cross-event information. These results validate the capability of handling EAE using token linking models.

Ablation Study of Two-fold Fusion To analyze the benefit of our proposed two-fold fusion, we conduct an ablation study based on RoBERTa-large. Table 4 illustrates that removing the First or Second-fold Fusion leads to a performance drop.

This suggests that both fusions contribute quite significantly to our model.

Case Study Here, we conduct qualitative analysis with a specific instance from WikiEvents. Figure 3 shows the EAE results from our Sep2F and SingleE. We can see that SingleE misses the two arguments in the "*Attack*" event triggered by "*booming*", but our Sep2F gives the correct predictions. We infer that our model can leverage the role information from the event triggered by "*killed*". Similarly, as the cross-event role "*Destroyer*" provides a vital clue, our model avoids the disturbance of the trigger "*death*" and recognizes the role of "*He*" as "*Killer*" rather than "*Victim*", but SingleE fails.

5 Conclusion

In this paper, we propose Sep2F, a novel multiple token linking model for EAE. Specifically, we employ two linking modules to separate the acquisition of cross-event information and the argument extraction of a target event. In addition, we propose a novel two-fold fusion module to guarantee that the acquired cross-event information enhances the argument extraction effectively. Therefore, the proposed model can leverage cross-event clues and retain the merits of single-event extraction. Extensive experiments on four widely used benchmarks show our model achieves new state-of-the-art performance.

Limitations

The limitations of our work are summarized as follows:

- We mainly focus on the EAE task in this paper. As multiple token linking models adapt to different information extraction tasks, such as event detection and relation extraction, we will extend our work and consider different designs for cross-event/cross-relation information acquisition in these tasks.
- How to leverage external resources in our proposed Sep2F remains an open question. The external resources can be other EAE datasets or commonsense knowledge and help enhance the EAE performance.

Ethics Considerations

Our work adheres to the guidelines outlined in the ACL Code of Ethics. As event argument extraction is a widely accepted and long-standing research task in NLP, we do not see any significant ethical concerns. As for the scientific artifacts used in our experiments, we confirm to comply with the corresponding intended use and licenses.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (Grant No. 2022YFC3302100) and the China Postdoctoral Science Foundation (Grant No. 2022TQ0033). This work was also supported by the financial support (No. 202206030113) by China Scholarship Council during a visit to Nanyang Technological University.

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.

George R 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)*.

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.

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.

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.

Yi Fung, Han Wang, Tong Wang, Ali Kebarighotbi, Mohit Bansal, Heng Ji, and Prem Natarajan. 2023. Deepmaven: Deep question answering on long-distance movie/tv show videos with multimedia knowledge extraction and synthesis. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3033–3043.

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

Yuxin He, Jingyue Hu, and Buzhou Tang. 2023. Revisiting event argument extraction: Can EAE models learn better when being aware of event co-occurrences? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12542–12556, Toronto, Canada. Association for Computational Linguistics.

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

I-Hung Hsu, Zhiyu Xie, Kuan-Hao Huang, Prem Natarajan, and Nanyun Peng. 2023. AMPERE: AMR-aware prefix for generation-based event argument extraction model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10976–10993, Toronto, Canada. Association for Computational Linguistics.

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

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

Hao Li, Yanan Cao, Yubing Ren, Fang Fang, Lanxue Zhang, Yingjie Li, and Shi Wang. 2023b. Intra-event and inter-event dependency-aware graph network for event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6362–6372, Singapore. 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.

Jiaju Lin and Qin Chen. 2021. Poke: A prompt-based knowledge eliciting approach for event argument extraction.

Chengyuan Liu, Fubang Zhao, Yangyang Kang, Jingyuan Zhang, Xiang Zhou, Changlong Sun, Kun Kuang, and Fei Wu. 2023a. RexUIE: A recursive method with explicit schema instructor for universal information extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15342–15359, Singapore. Association for Computational Linguistics.

Fuxiao Liu, Yaser Yacoob, and Abhinav Shrivastava. 2023b. Covid-vts: Fact extraction and verification on short video platforms. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 178–188.

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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2023. Universal information extraction as unified semantic matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13318–13326.

Jie Ma, Shuai Wang, Rishita Anubhai, Miguel Ballesteros, and Yaser Al-Onaizan. 2020. Resource-enhanced neural model for event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3554–3559.

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.

Chien Nguyen, Hieu Man, and Thien Nguyen. 2023. Contextualized soft prompts for extraction of event arguments. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4352–4361, Toronto, Canada. Association for Computational Linguistics.

OpenAI. 2022. Introducing chatgpt. `https://openai.com/blog/chatgpt`.

Sampo Pyysalo, Tomoko Ohta, Makoto Miwa, Han-Cheol Cho, Jun'ichi Tsujii, and Sophia Ananiadou. 2012. Event extraction across multiple levels of biological organization. *Bioinformatics*, 28(18):i575–i581.

Jianlin Su. 2019. Conditional text generation based on conditional layer normalization. `https://spaces.ac.cn/archives/7124`.

Wei Tang, Benfeng Xu, Yuyue Zhao, Zhendong Mao, Yifeng Liu, Yong Liao, and Haiyong Xie. 2022. Unirel: Unified representation and interaction for joint relational triple extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7087–7099.

Hai-Long Trieu, Thy Thy Tran, Khoa NA Duong, Anh Nguyen, Makoto Miwa, and Sophia Ananiadou. 2020. Deepeventmine: end-to-end neural nested event extraction from biomedical texts. *Bioinformatics*, 36(19):4910–4917.

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.

Sijia Wang, Mo Yu, Shiyu Chang, Lichao Sun, and Lifu Huang. 2022. Query and extract: Refining event extraction as type-oriented binary decoding. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 169–182.

Xiaozhi Wang, Ziqi Wang, Xu Han, Zhiyuan Liu, Juanzi Li, Peng Li, Maosong Sun, Jie Zhou, and Xiang Ren. 2019. Hmeae: Hierarchical modular event argument extraction. In *Proceedings of the 2019 Conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 5777–5783.

Kaiwen Wei, Xian Sun, Zequn Zhang, Jingyuan Zhang, Guo Zhi, and Li Jin. 2021. Trigger is not sufficient: Exploiting frame-aware knowledge for implicit event argument 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 4672–4682.

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

Haoyang Wen, Ying Lin, Tuan Lai, Xiaoman Pan, Sha Li, Xudong Lin, Ben Zhou, Manling Li, Haoyu Wang, Hongming Zhang, et al. 2021. Resin: A dockerized schema-guided cross-document cross-lingual cross-media information extraction and event tracking system. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 133–143.

David H Wolpert. 1992. Stacked generalization. *Neural networks*, 5(2):241–259.

Xueqing Wu, Kung-Hsiang Huang, Yi Fung, and Heng Ji. 2022. Cross-document misinformation detection based on event graph reasoning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 543–558.

Xi Xiangyu, Wei Ye, Shikun Zhang, Quanxiu Wang, Huixing Jiang, and Wei Wu. 2021. Capturing event argument interaction via a bi-directional entity-level recurrent decoder. 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 210–219.

Jun Xu, Weidi Xu, Mengshu Sun, Taifeng Wang, and Wei Chu. 2022a. Extracting trigger-sharing events via an event matrix. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1189–1201.

Runxin Xu, Peiyi Wang, Tianyu Liu, Shuang Zeng, Baobao Chang, and Zhifang Sui. 2022b. A two-stream amr-enhanced model for document-level event argument extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5025–5036.

Bowen Yu, Zhenyu Zhang, Jiawei Sheng, Tingwen Liu, Yubin Wang, Yucheng Wang, and Bin Wang. 2021. Semi-open information extraction. In *Proceedings of the Web Conference 2021*, pages 1661–1672.

Qi Zeng, Qiusi Zhan, and Heng Ji. 2022. Ea2e: Improving consistency with event awareness for document-level argument extraction. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2649–2655.

Kaihang Zhang, Kai Shuang, Xinyue Yang, Xuyang Yao, and Jinyu Guo. 2023. What is overlap knowledge in event argument extraction? APE: A cross-datasets transfer learning model for EAE. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 393–409, Toronto, Canada. Association for Computational Linguistics.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022. Transfer learning from semantic role labeling to event argument extraction with template-based slot querying. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2627–2647.

Jie Zhou, Qi Zhang, Qin Chen, Liang He, and Xuan-Jing 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*, pages 1990–2000.

A Dataset Details

ACE05 (Doddington et al., 2004) is a sentence-level information extraction dataset including entities, relations and events with three language

Hyperparameter	Searching Interval	ACE05	RAMS	WikiEvents	MLEE
Batch Size	-	16	8	8	8
Training Epoch	-	100	30	100	100
Learning Rate	[3e-5, 4e-5, 5e-5]	3e-5	5e-5	4e-5	4e-5
Training Weight α	[0.5, 0.6, 0.7, 0.8, 0.9]	0.7	0.7	0.8	0.6
Projection Dimension d_2	[32, 64]	32	64	64	64
Batch Size	-	16	8	8	8
Training Epoch	-	100	50	100	100
Learning Rate	[1e-5, 2e-5, 3e-5]	2e-5	3e-5	2e-5	1e-5
Training Weight α	[0.5, 0.6, 0.7, 0.8, 0.9]	0.8	0.5	0.7	0.8
Projection Dimension d_2	[32, 64]	32	64	64	64

Table 5: Hyperparameter settings. The upper table shows the setting details of our base model, while the bottom table corresponds to our large model.

versions. It consists of annotated newspapers, newswire data and broadcast news through the efforts of the Automatic Content Extraction (ACE) program. We use its English event annotation as our evaluation for the sentence-level EAE. As for the data preprocessing, we utilize the scripts of EEQA (Du and Cardie, 2020), which follows the settings of DyGIE++ (Wadden et al., 2019).

RAMS (Ebner et al., 2020) is a document-level EAE dataset derived from news articles. Unlike the original annotations, which treat multiple events in the same context as different instances, we follow the preprocessing procedure of TabEAE (He et al., 2023) to aggregate the annotations of multiple events appearing within the same context for each instance. The aggregation setting does not bring additional resources or knowledge. Furthermore, we continue to extract one target event from each instance individually, and the number of instances in RAMS remains unchanged.

WikiEvents (Li et al., 2021) is a document-level EAE dataset sourced from English Wikipedia. We follow the preprocessing procedure of PAIE (Ma et al., 2022) to employ a pre-defined window centering on each trigger word to avoid exceeding the length constraint of PLMs. It differs from the window setting of TabEAE and facilitates the better utilization of multiple events when keeping the same length of the window.

MLEE (Pyysalo et al., 2012) is a document-level event extraction dataset annotated from the abstracts of English publications in the biomedical field. We follow the preprocessing procedure of TabEAE, which refers to the work (Trieu et al., 2020). Besides, as no development set in MLEE, we follow TabEAE to use the training set to tune our hyperparameters.

Dataset	ACE05	RAMS	WikiEvents	MLEE
#Events				
Train	4,202	7,329	3,241	4,442
Dev	450	924	345	-
Test	403	871	365	2,200
#Args				
Train	4,859	17,026	4,552	5,786
Dev	605	2,188	428	-
Test	576	2,023	566	2,764
#Event Types	33	139	50	23
#Role Types	22	65	59	8
#Avg Args	1.19	2.33	1.40	1.29

Table 6: Detailed statistics of datasets. *Avg Args* denotes the average number of arguments per event.

Statistics Table 6 lists the detailed statistics of the above four datasets.

B Implementation Details

Following TabEAE, we set the pre-defined window length as 250 on all four datasets. For the training epoch, as our source code refers to the implementation of UniRel (Tang et al., 2022), we keep its training epoch settings except for RAMS. A smaller training epoch is chosen because RAMS contains more training instances than the other three datasets. In particular, we search the training epoch within the interval [30, 50] for RAMS. For the batch size, we set a maximum value (a power of 2) that a single GPU can run on the three document-level datasets. For the sentence-level ACE05, we search the batch size within the interval [8, 16]. Note that we first tune the learning rate and the training weight α by a grid search based on the development set of each dataset. After that, we keep these two hyperparameter settings and tune the other hyperparameters. The tuned intervals and chosen hyperparameters are presented in Table 5.

α	ACE05	RAMS	WikiEvents	MLEE
0.5	76.1	**53.7**	68.6	76.6
0.6	76.8	53.0	68.7	76.4
0.7	76.7	53.1	**69.3**	76.6
0.8	**77.0**	52.7	**69.3**	**76.7**
0.9	76.4	52.8	67.9	76.4

Table 7: Arg-C F1 (%) results with different training weight hyperparameters. The performance is reported based on RoBERTa-large.

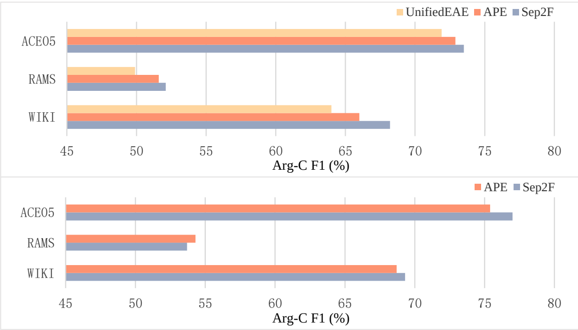

Figure 4: Performance comparison with the models trained on multiple datasets. The upper figure illustrates the comparison using base-version PLMs, while the bottom figure shows the results with large-version PLMs.

C Analysis on Weight Hyperparameters

For the weight hyperparameter α, we maintain the chosen learning rate of each dataset and analyze the performance in Arg-C F1 (%) when tuning it within the interval [0.5, 0.6, 0.7, 0.8, 0.9]. As shown in Table 7, we observe that our Sep2F exhibits relatively stable performance across different values of α. This proves the robustness of Sep2F well. Additionally, when we set α to 1, it implies that we are unable to utilize the cross-event role labels by tuning the trigger-role loss $\mathcal{L}_{\text{TR}}$ in Equation (3). As a result, the absence of cross-event role information degrades the performance of our model in Arg-C F1 by 1.2%, 1.8%, 2.3% and 1.9% on ACE05, RAMS, WikiEvents and MLEE, respectively.

D Prompt Construction for ChatGPT

To evaluate the EAE performance of ChatGPT, we follow Han et al. (2023) to construct 5-shot in-context learning (ICL) (Brown et al., 2020) prompts. Each constructed prompt consists of three components: Instruction, Demonstration and Target. The details of them are listed as follows:

Instruction describes EAE and specifies the output format. We use the task instruction and output format description provided by Han et al. (2023).

Instruction	Based on the given text and an event it involved, first find out all arguments of this event from the given text, then assign a role to each argument from the given candidate roles. The argument is an entity that appears in the given text and participates in this event. Since the arguments in the given text may come from multiple events, please identify only the arguments of the given event. Answer in the format ["argument", "role"] without any explanation. If no argument is involved, then just answer "[]".
Demonstration	Given text: Belgium Police Release New Video of Brussels Bombing Suspect PARIS—Belgian police have released new video of a wanted Brussels airport bombing suspect even as the lawyer for a Paris attacks suspect says his extradition may take a few more weeks. Released Thursday in French and Flemish, the police video shows the minutes following the March 22 Zaventem Airport bombings and the apparent getaway of the third surviving suspect--often identified in the media as "the man with the hat". Local media previously released security camera video of the man at Zaventem Airport shortly before the bombings, wearing a hat and a light jacket and walking alongside suicide bombers Ibrahim el-Bakraoui and Najim Laachraoui. All three men are seen pushing carts with bags on them. With running commentary in French and Flemish, the police footage shows the third man leaving the airport after the bombs went off at 7:58 a.m. First he is walking, then he breaks into a jog. His face is not seen clearly in the new images. Event trigger: "bombings" in "the bombings ," Event type: "Conflict.Attack.DetonateExplode" Candidate roles: ["Attacker", "ExplosiveDevice", "Instrument", "Place", "Target"] Answer: [["Zaventem Airport", "Place"]] Given text: ......
Target	Given text: Two senior officials in the interior ministry said the exact casualty figures were not being disclosed to prevent unrest within the armed forces. "I have been told not to make the death toll figures public. It is frustrating to hide the facts," said a senior interior ministry official in Kabul. A senior NDS official in Kabul said at least 50 people were killed or wounded in the complex attack. Abdurrahman Mangal, spokesman for the provincial governor in Maidan Wardak said 12 people were killed and 12 were injured when the car bomb exploded near the Afghan special forces unit. President Ashraf Ghani's office in a statement said the "enemies of the country" had carried out an attack against NDS personnel in Maidan Shahr. "They killed and wounded a number of our beloved and honest sons." Turkish President Recep Tayyip Erdoğan condemned the attack on Monday evening and extended his condolences to Ghani. Meanwhile, the Taliban said it met with U.S. officials in Qatar on Monday, in the latest round of talks between the insurgents and Washington aimed at bringing an end to the 17-year war. The U.S. has not officially commented on the reported meeting, which follows the last confirmed talks between the two parties in the UAE in December. Event trigger: "killed" in "They killed and" Event type: "Life.Die.Unspecified" Candidate roles: ["Killer", "Place", "Victim"] Answer:
Output	[["They", "Killer"], ["beloved and honest sons", "Victim"]]

Figure 5: A prompt example from WikiEvents.

Demonstration contains five randomly sampled training instances. Each sampled training instance includes input text, event trigger information, event type information, event-specific candidate roles and golden argument extraction results. Note that we also provide a short context centering on the trigger in the trigger information. It helps ChatGPT locate the trigger from the possible repeated words.

Target refers to the test instance. We provide its input text, event trigger, event type and event-specific candidate roles.

Then, we concatenate and feed the above three parts into ChatGPT to acquire the EAE results for each test instance. Specifically, ChatGPT is expected to output an argument list and each argument consists of its text span and role type. A prompt example is shown in Figure 5.

E Comparison with Resource-enhanced Models

We compare our model with UnifiedEAE (Zhou et al., 2022) and APE (Zhang et al., 2023), which focus on exploring cross-dataset knowledge. In particular, APE pays extra manual efforts to design overlap knowledge prompts. From the results in Figure 4, we observe that our Sep2F outperforms UnifiedEAE and APE on all three datasets when using base-version PLMs. Furthermore, our Sep2F performs better than APE on ACE05 and WikiEvents and obtains fairly competitive performance on RAMS when using large-version PLMs. Thus, we summarize that our model achieves the best performance on almost all datasets utilizing different versions of PLMs, though the compared models benefit from additional training resources and manual efforts. The promising results further prove the superiority of our model.