Fusion meets Function: The Adaptive Selection-Generation Approach in Event Argument Extraction

Guoxuan Ding[1,2], Xiaobo Guo[1], Xin Wang[1], Lei Wang[1], Tianshu Fu[1*], Nan Mu[1], Daren Zha[1]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China,
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{dingguoxuan,guoxiaobo,wangxin,wanglei,futianshu,munan,zhadaren}@iie.ac.cn

Abstract

Event Argument Extraction is a critical task of Event Extraction, focused on identifying event arguments within text. This paper presents a novel Fusion Selection-Generation-Based Approach, by combining the precision of selective methods with the semantic generation capability of generative methods to enhance argument extraction accuracy. This synergistic integration, achieved through *fusion prompt*, *element-based extraction*, and *fusion learning*, addresses the challenges of input, process, and output fusion, effectively blending the unique characteristics of both methods into a cohesive model. Comprehensive evaluations on the RAMS and WIKIEVENTS demonstrate the model's competitive performance and efficiency.

1 Introduction

Event Argument Extraction (EAE) is a critical subtask in the field of Event Extraction, aiming to identify arguments of known events in text (Li et al., 2022). For instance, in the sentence "Bryant debated against Torres's statement that tax reforms were not benefitting the middle class in Florida." Here, *debated* serves as the trigger word, indicating a CONTACT.NEGOTIATE event. This event involves multiple arguments such as *Bryant* (participant), *Torres* (participant), *tax reforms* (topic), and *Florida* (place), with the terms in parentheses representing their respective argument roles. The challenge of EAE lies in accurately extracting corresponding event arguments from texts under a given event theme.

Traditional EAE methods are based on selective models that focused on recognizing or tagging existing elements or patterns in texts (Yang et al., 2019a), such as the paradigm of Sequence Labeling and Token Classification (Wang et al., 2020; Lu et al., 2021; Shi and Lin, 2019). While these methods utilize model structural complexity to adapt to training data, their reliance on rote learning limit their effectiveness in leveraging semantic information, thus constraining their ability to uncover unknown knowledge.

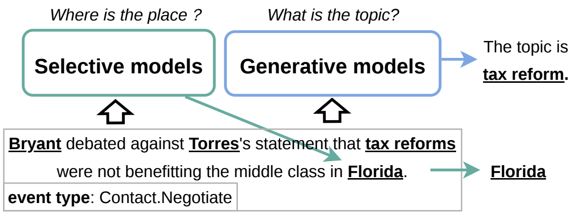


Figure 1: Difference Between Selective and Generative Methods in Event Argument Extraction: Selective methods identify tokens in text for answer selection, while generative methods employ natural language generation to produce exhaustive answer sequences.

The advent of Pre-Trained Models (PTMs) has led to a paradigm shift (Sun et al., 2022) in EAE, emphasizing their advanced text generation and semantic understanding. This shift is marked by a move from traditional selective methods to Machine Reading Comprehension (Du and Cardie, 2020; Ma et al., 2022; Liu et al., 2020) for extracting answer spans within text through question formulation. Simultaneously, there is a growing inclination towards generative methods, exemplified by the Sequence-to-Sequence paradigm (Lu et al., 2021; Li et al., 2021), which redefines EAE as a sequence generation task.

Despite this progress, both selective and generative methods continue to face distinct challenges. Selective methods, while precise and mature in identifying specific text elements, often fall short in deep semantic processing, a critical aspect for comprehending intricate textual nuances. In contrast, generative methods excel in producing detailed and nuanced outputs, but face hurdles in accurately extracting pertinent information from their

[*]Corresponding author

Proceedings of the 31st International Conference on Computational Linguistics, pages 4359–4369
January 19–24, 2025.

extensive, generated sequences. These challenges highlight the need for balanced methodologies in EAE, where the complementary strengths of selection and generation methods work together to enhance model performance.

Since both methods exhibit distinct input, process, and output characteristics, we focus on the following key questions to explore their integration:

- How can we effectively blend the unique input characteristics of both methods within the fusion model?
- In what ways can the distinct processing techniques of each method be integrated to optimize the overall workflow?
- What methods can be employed to harmonize the differing output formats of these methods within the unified model framework?

In this paper, we introduce a Fusion Selection-Generation-Based Approach for EAE, synergizing the capabilities of both selective and generative methods. To solve the questions mentioned above, we propose three key technologies: diversified element fusion prompts, independent element-based extraction parts and a cohesive fusion learning process.

The fusion prompt is designed as an integrative structure, encompassing various elements such as argument roles, event arguments, and masks, each aligning with specific extraction parts. Element-based extraction parts comprises two distinct components: the selection part, which utilizes argument roles combined with trigger knowledge for precise identification of argument positions; and the generation part, which prompts the model to generate argument sequences based on event arguments. These two parts are trained in parallel, each with distinct loss calculations. This dual-learning fosters an initial integration of the methodologies within model. As training progresses, a dynamic masking mechanism within the fusion prompts subtly incorporates the generative part into the selection framework. This fusion learning process leads to a unified approach in loss computation, harmonizing the strengths of both parts to ensure consistent and optimized outcomes.

Our experiments on the RAMS and WIKIEVENTS demonstrate that our fusion model achieves competitive performance, excelling in various metrics while also showcasing superior efficiency, including enhanced extraction efficiency and reduced memory usage. To study how our model functions, we delve into analyses based on ablation studies and fusion strategies, uncovering insights into the effective integration of selection and generation parts. This comprehensive approach reinforces our model's adaptability in the field of EAE.

In summary, the main contributions of this paper include:

- We present a novel Fusion Selection-Generation-Based Approach that effectively combines the strengths of both selective and generative methods, enhancing the accuracy of event argument extraction.
- Our model utilizes fusion prompts and a fusion learning process to promote the fusion of two distinct element-based extraction parts, encompassing both selection and generation.
- The fusion model achieves competitive performance, showcasing high accuracy and efficiency. We present in-depth analyses of ablation studies and fusion strategies to demonstrate the model's effective integration.

2 Methodology

2.1 Problem Formulation

For an EAE task, consider a text collection $\mathcal{T} = \{X_i | i = 1, \ldots, |\mathcal{T}|\}$ and a set of event types $\mathcal{E} = \{e_i | i = 1, \ldots, |\mathcal{E}|\}$. Each sample X is a token sequence that corresponds to an event type e, along with the trigger x_t in the text for that event. Each event type is associated with a set of argument roles R, and an event theme is defined by an event type e and its corresponding R_e. The task of EAE involves extracting all (r, a) pairs from the text X under a given event theme, where r in R is an argument role in the event theme, and a is the event argument corresponding to r, specifically the text span in X.

Given an event type and its associated argument roles, we can create a corresponding prompt Pt (Qin and Eisner, 2021; Liu et al., 2023). To leverage the generative model's capabilities, we use a Manual Template (Li et al., 2021), closely resembling natural language, for input concatenation in the encoder and integration in the decoder.

2.2 Model Overview

In this paper, we introduce a novel Fusion Selection-Generation-Based Approach. Our model comprises three connected components: *fusion prompt*, *element-based extraction*, and *fusion learning*. Central to this approach is *fusion prompt*, in which elements guide the computations in *element-based extraction* containing both the selection and generation parts.

As depicted in Figure 3a, the selection and generation parts operate independently and simultaneously. The selection part focuses on accurately identifying argument positions within the text, while the generation part is tasked with creating the corresponding argument sequences. These two processes, though distinct, are seamlessly integrated through *fusion learning*, as illustrated in Figure 3b. This integration unifies the training procedure and harmonizes the outputs of both components.

2.3 Preparations

The model adopts an encoder-decoder architecture (*e.g.*, BART (Lewis et al., 2020), T5 (Raffel et al., 2020)). We concatenate the text and original prompt as input X_{pt} to the encoder, represented as <s> $\tilde{X}$ [SEP] Pt </s>, where $\tilde{X}$ is the text X annotated with the trigger, *i.e.*, $\tilde{X} = [\ldots, x, \text{<t>}, x_t, \text{</t>}, \ldots]$. The encoder's autoencoder model (Vaswani et al., 2017) facilitates comprehensive self-attention computation on the input, effectively embedding the text with prompt-derived role information. We obtain the text representation infused with prompt information from the encoder:

$$\boldsymbol{H_T} = \text{Encoder}(X_{pt}) \tag{1}$$

The prompt Pt is designed as a versatile fusion structure, capable of incorporating various elements such as argument roles r, event arguments a, argument masks <$mask$>, and natural language connectives:

... with ***participant*** about ... (original role)

... with ***participant*** **Torres** about ... (role with argument)

... with ***participant*** **<mask>** about ... (role with mask)

The entire fusion process is illustrated in Figure 2. We utilize the fused Pt as the input for the decoder, whose autoregressive model (Yang et al., 2019b) with strong generative capabilities aids in processing the prompt to generate event arguments within Pt. We obtain the fusion prompt representation from the decoder:

$$\boldsymbol{H_{Pt}} = \text{Decoder}(Pt; \boldsymbol{H_T}) \tag{2}$$

To describe the process of obtaining targeted element representation from $\boldsymbol{H_T}$ and $\boldsymbol{H_{Pt}}$, we introduce the following definition: Given the textual element x_e, with its start position i and end position j in the input X, the formula is given by:

$$\boldsymbol{X_e} = \text{Retrieve}(x_e; \boldsymbol{H}) \tag{3}$$

where $\boldsymbol{X_e}$ represents the embedding vectors corresponding to positions i to j in $\boldsymbol{H}$, the output of the encoder or decoder, based on the input X.

2.4 Selection Part

In the Selection Part, our approach embraces the MRC paradigm, focusing on pinpointing the start and end positions of arguments within the text. The method facilitates the interaction between role representations and their corresponding textual contexts. This process leverages the combined strengths of both the encoder and the decoder to acquire comprehensive representations:

$$\begin{aligned} \boldsymbol{H_X} &= \text{Retrieve}(\tilde{X}; \boldsymbol{H_T}) \in \mathbb{R}^{h \times L} \\ \boldsymbol{h_r} &= \text{Retrieve}(r; \boldsymbol{H_{Pt}}) \in \mathbb{R}^{h} \\ \boldsymbol{h_t} &= \text{Retrieve}(x_t; \boldsymbol{H_T}) \in \mathbb{R}^{h} \end{aligned} \tag{4}$$

where h and l respectively represent the dimension of hidden layer and the maximum length of the text, and r encapsulates all the argument roles within Pt. In instances where the textual elements r and x_t comprise multiple tokens, the representations of these tokens are averaged to form a unified vector respectively, encapsulating the collective characteristics of all tokens within the textual element.

To integrate more text information, we adopt an embedding interactions method (Zhou et al., 2020) to merge triggers into the roles, denoted by $\boldsymbol{h_{r,t}} = [\boldsymbol{h_r}, \boldsymbol{h_t}, \boldsymbol{h_r} \odot \boldsymbol{h_t}, \boldsymbol{h_r} - \boldsymbol{h_t}]$, where $[\cdot, \cdot]$ denotes a vector concatenation, and $\odot$ is the element-wise Hadamard product. Then this fusion representation undergoes attention computation with the text representations, resulting in text-sized probability distribution of positions:

$$\begin{aligned} \boldsymbol{p}^{(\text{sel\_start})} &= \text{Softmax}(\boldsymbol{h}_{\boldsymbol{r,t}}^T \boldsymbol{V_s} \boldsymbol{H_X}) \in \mathbb{R}^L \\ \boldsymbol{p}^{(\text{sel\_end})} &= \text{Softmax}(\boldsymbol{h}_{\boldsymbol{r,t}}^T \boldsymbol{V_e} \boldsymbol{H_X}) \in \mathbb{R}^L \end{aligned} \tag{5}$$

where $\boldsymbol{V_s}, \boldsymbol{V_e} \in \mathbb{R}^{4h \times h}$ are learnable parameter matrices shared across all roles. They encapsulate

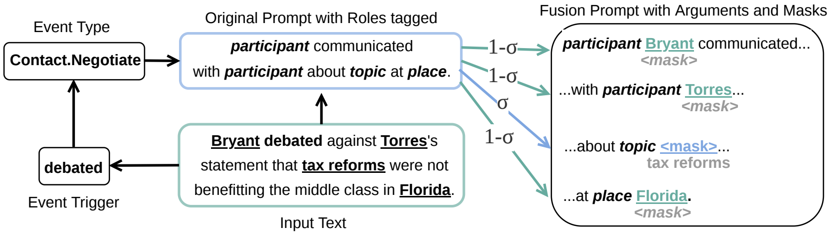
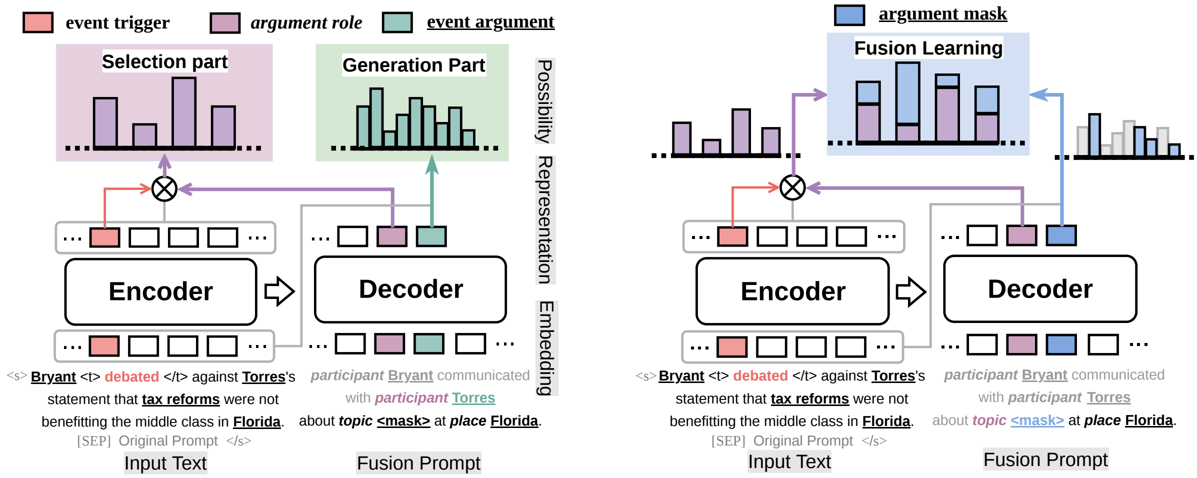


Figure 2: *Fusion Prompt*: Given an input text and its event type, the original prompt is obtained. Under the current probability σ, event arguments and masks are randomly integrated after argument roles, creating the final fused Pt.



(a) Selection and Generation Parts are trained concurrently, utilizing roles and arguments from the fusion prompt separately.

(b) Selection and Generation Parts are integrated by using <*mask*> from the fusion prompt and a dynamic masking mechanism.

Figure 3: Learning Process of the Fusion Model: (a) *Element-based Extraction*: For the currently queried argument roles, when followed by event arguments, they are trained separately using selection and generation parts, primarily in the initial stages of model training. (b) *Fusion Learning*: When the queried training roles are followed by masks, the generation part leverages learned knowledge to align with the the selection part, facilitating their integration.

key information about the argument positions of roles.

For the current query role k, we have a selective loss function:

$$\mathcal{L}_{\text{sel}}(k) = -\big(s_k \log \boldsymbol{p}_{\boldsymbol{k}}^{(\text{sel\_start})} + e_k \log \boldsymbol{p}_{\boldsymbol{k}}^{(\text{sel\_end})}\big) \tag{6}$$

where s_k, e_k represent the true labels for the start and end positions of the argument span in text for the current role.

2.5 Generation Part

In the Generation Part, we adopt a specialized generation technique inspired by BART-Gen (Li et al., 2021), which leverages the decoder's hidden layers along with text embeddings to generate a vocabulary-sized probability distribution. Distinct from BART-Gen, our method is specifically tailored to bypass the generation of complete sentences. Instead, it concentrates on learning and accurately generating the specific event arguments, thereby refining the efficiency of the extraction process.

To accurately represent the event arguments in our model, we utilize the decoder to derive their representations and apply the embedding layer to encode the tokens from the input text. Specifically, the process is defined as follows:

$$\begin{aligned} \boldsymbol{H_a} &= \text{Retrieve}(a; \boldsymbol{H_{Pt}}) \in \mathbb{R}^{h \times d} \\ \boldsymbol{E_X} &= \text{Embedding}(\tilde{X}) \in \mathbb{R}^{h \times L} \end{aligned} \tag{7}$$

Here, a encapsulates all the event arguments within $\tilde{X}$. Considering that a might comprise multiple tokens $[a_1, \ldots, a_d]$, the representation $\boldsymbol{H_a}$ is thus a sequence of token embeddings, with each $\boldsymbol{h_a^i} \in \mathbb{R}^h$ serving as the representation of the token a_i. In line with the BART-Gen approach, we perform a dot product operation between $\boldsymbol{h_a}$ and $\boldsymbol{E_X}$, generating an initial probability distribution. To extend this distribution to encompass the full vocabulary, we append zeros for the vocabulary words absent in the text X. For each token a_i in a, the probability distribution is given by:

$$\boldsymbol{p}_{\boldsymbol{a_i}}^{(\text{vocab})} = \begin{cases} \boldsymbol{h_a^i}^T \text{Retrieve}(w; \boldsymbol{E_X}), & w \in X \\ 0, & w \notin X \end{cases} \tag{8}$$

where w denotes every word in the vocabulary.

For the current role k, we have a generative loss function:

$$\mathcal{L}_{\text{gen}}(k) = -\sum_{k_i=1}^{d} v_k \log \boldsymbol{p}_{\boldsymbol{k_i}}^{(\text{vocab})} \tag{9}$$

where v_k indicates the true label for the position of the current event argument in the vocabulary.

2.6 Fusion Learning

In the Fusion Learning, our model employs a dynamic masking mechanism to integrate selection and generation parts. Throughout the training process, event arguments within Pt are randomly masked, with the likelihood of this masking operation increasing incrementally across training iterations. This probability, denoted as σ, is defined as:

$$\sigma = \frac{\text{current training times}}{\text{max training times}} \in (0, 1) \tag{10}$$

For each role r, the corresponding event argument a is retained with a probability of $1 - \sigma$ for generation part, while the mask token <$mask$> is applied with a probability of σ, thereby facilitating the integration with the selection part. The representation of <$mask$>, derived from the decoder, is given by:

$$\boldsymbol{h_m} = \text{Retrieve}(m; \boldsymbol{H_{Pt}}) \in \mathbb{R}^h \tag{11}$$

where m represents the <$mask$> following the role. To achieve fusion of both parts, we adopt a loss function analogous to that used in the selection part:

$$\begin{aligned} \boldsymbol{p}^{(\text{gen\_start})} &= \text{Softmax}(\boldsymbol{h_m^T E_X} \odot \boldsymbol{w_s}) \in \mathbb{R}^L \\ \boldsymbol{p}^{(\text{gen\_end})} &= \text{Softmax}(\boldsymbol{h_m^T E_X} \odot \boldsymbol{w_e}) \in \mathbb{R}^L \end{aligned} \tag{12}$$

where $\boldsymbol{w_s}, \boldsymbol{w_e} \in \mathbb{R}^L$ are learnable parameter vectors.

For the current role k, we define a mask loss function:

$$\begin{aligned} \mathcal{L}_{\text{msk}}(k) = -\big(s_k \log \boldsymbol{p}_{\boldsymbol{k}}^{(\text{gen\_start})} \\ + e_k \log \boldsymbol{p}_{\boldsymbol{k}}^{(\text{gen\_end})}\big) \end{aligned} \tag{13}$$

where s_k and e_k serve the same roles as in $\mathcal{L}_{\text{sel}}$. Subsequently, to achieve an effective balance between the selection and generation parts within our fusion model, we compute the fusion loss function as:

$$\mathcal{L}_{\text{fus}}(k) = \lambda \mathcal{L}_{\text{sel}}(k) + (1 - \lambda)\mathcal{L}_{\text{msk}}(k) \tag{14}$$

where fusion ratio λ is a weighting factor that modulates the contribution of selection and generation parts to the overall fusion loss, optimizing the synergy between these two parts for enhanced model performance.

2.7 Overall Loss

For the current sample t, let R_t be the set of roles corresponding to the event in this sample. In the current training iteration, n roles have their corresponding arguments replaced by <$mask$>, forming the subset R_{mask}. The overall loss function for the current sample is given by:

$$\begin{aligned} \mathcal{L} = &\sum_{k \in (R_t - R_{mask})} (\mathcal{L}_{\text{sel}}(k) + \mathcal{L}_{\text{gen}}(k)) \\ &+ \sum_{k \in R_{mask}} \mathcal{L}_{\text{fus}}(k) \end{aligned} \tag{15}$$

During the testing phase, we use a prompt where all the arguments corresponding to the roles are masked. For any role k in the current sample, its start and end positions are computed as follows:

$$\begin{aligned} k_{\text{start}} &= \arg\max(\lambda p_k^{\text{sel\_start}} + (1 - \lambda) p_k^{\text{gen\_start}}) \\ k_{\text{end}} &= \arg\max(\lambda p_k^{\text{sel\_end}} + (1 - \lambda) p_k^{\text{gen\_end}}) \end{aligned} \tag{16}$$

3 Experiments

3.1 Setup

Datasets We utilize two document-level event argument extraction datasets: RAMS (Ebner et al., 2020) and WIKIEVENTS (Li et al., 2021). RAMS comprises 9,124 news examples with 139 event types and 65 argument roles. WIKIEVENTS, extracted from English Wikipedia articles, includes 246 documents with 50 event types and 59 argument roles. The detailed statistics of two datasets are listed in Appendix A.1.

Evaluation Metrics In evaluating our model, we adopt F1 score as the key metric across three primary aspects: Argument Identification (Arg-I), Argument Classification (Arg-C), and Head Classification (Head-C):

- **Arg-I:** Argument Identification focuses on the accurate prediction of offsets for any given role's event arguments.

- **Arg-C:** Argument Classification involves correctly identifying both the position and type of argument roles.

- **Head-C:** Specifically used for WikiEvent (Li et al., 2021), Head Classification assesses the accuracy of predicting the headwords of arguments.

Each of these metrics plays a crucial role in assessing the overall performance of our model, offering a comprehensive view of its capabilities in various dimensions of EAE.

Baselines We assess the performance of our model against a range of established models in EAE: (1) Selective Models: BERT-CRF (Shi and Lin, 2019), EEQA (Du and Cardie, 2020) and PAIE (Ma et al., 2022). (2) Generative Models: BART-Gen (Li et al., 2021) and Retrieval-augmented (Ren et al., 2023). These baseline models are selected to represent both selective and generative methods, providing a comprehensive overview of current EAE techniques. The detailed of these models are outlined in Appendix A.2.

Experimental Configuration Our experiments leverage the encoder-decoder architecture of the pretrained BART model, obtained in two model sizes, base and large, containing respectively 139M and 406M parameters, from the Hugging Face repository[1]. This choice is guided by our intention to investigate the impact of model size on performance in our fusion model. We do not use concatenated input text on the RAMS. For each training iteration, we use random seeds [13, 21, 44, 88, 100] and three learning rates [2e-5, 3e-5, 5e-5]. The highest learning rate result for each seed is averaged to produce the final training result (Ren et al., 2023). We list other important hyperparameters in Appendix A.3.

[1] https://github.com/huggingface/transformers

To investigate the relative impact of selection and generation parts within fusion model, we implement three fusion model configurations: Fusion Generatively Biased, Fusion Balanced, and Fusion Selectively Biased, corresponding to fusion ratios λ of 0.2, 0.5, and 0.8. This experimental design allows us to systematically explore how varying degrees of bias towards either selection or generation parts influence the overall performance and characteristics of the model in EAE tasks.

3.2 Overall Performance

Table 1 presents the performance of all baselines and fusion models on RAMS and WikiEvents. From the results, we can conclude that:

(1) Compared to selective models, *the fusion model demonstrates considerable competitiveness, achieving strong performance with efficient resource usage.* On the WikiEvents, for instance, our model demonstrates exceptional performance, securing best and second best results in Head-C. This trend of excellence is mirrored in both the RAMS and WikiEvents, where our model achieves SOTA results in Arg-C.

While the improvement in metrics may not appear overwhelming at first glance, the model demonstrates clear advantages in both extraction efficiency and computational efficiency:

(i) Extraction efficiency, reflecting in the ratio of argument classification to identification, underscores the model's ability to minimize unnecessary span identifications while maintaining a balanced performance across both metrics. As shown in Table 2, on the BART-base, it achieves the best and second best results, while on BART-large, it surpasses the PAIE by 0.9% and 0.7%, *indicating a more efficient use of the model's inherent semantic capabilities*.

(ii) Moreover, our fusion model showcases a distinct advantage in terms of computational efficiency. The PAIE model, for example, necessitates the use of an encoder once and a decoder twice, leading to substantially higher memory consumption. In contrast, our fusion approach utilizes the complete encoder-decoder components only once, resulting in a more streamlined and resource-efficient process. To illustrate, as shown in Table 3, the PAIE model requires 136% of the GPU memory needed by our fusion model. This comparison highlights *our model's ability to deliver comparable or superior performance while significantly reducing computational load and memory usage*.

Models	RAMS		WikiEvents			PLM
	Arg-I	Arg-C	Arg-I	Arg-C	Head-C	
Selective Models						
BERT-CRF (Shi and Lin, 2019)*	-	40.3	-	32.3	43.3	BERT-base
EEQA (Du and Cardie, 2020)*	46.4	44.0	54.3	53.2	56.9	BERT-base
	48.7	46.7	56.9	54.5	59.3	BERT-large
PAIE (Ma et al., 2022)*	54.7	49.5	68.9	63.4	66.5	BART-base
	<u>56.8</u>	52.2	**70.5**	<u>65.3</u>	68.4	BART-large
Generative Models						
BART-Gen (Li et al., 2021)*	50.9	44.9	47.5	41.7	44.2	BART-base
	51.2	47.1	66.8	62.4	65.4	BART-large
Retrieval-augmented (Ren et al., 2023)*	53.3	46.3	61.4	46.1	62.5	T5-base
	54.6	48.4	69.6	63.4	68.4	T5-large
Fusion Selection-Generation-Based Models						
Fusion Generatively Biased	53.0	47.8	68.7	63.7	67.8	BART-base
	56.1	51.7	<u>70.1</u>	**65.4**	<u>68.5</u>	BART-large
Fusion Balanced	53.6	48.6	68.3	63.9	67.7	BART-base
	56.6	<u>52.5</u>	69.9	64.7	**68.8**	BART-large
Fusion Selectively Biased	53.5	48.4	68.7	63.3	67.5	BART-base
	56.9	**52.6**	69.9	64.4	68.1	BART-large

Table 1: Performance (%) of Arg-I and Arg-C on the RAM and WIKIEVENTS. * means the results from Ren et al. (2023). **Best results** are marked in bold, and the <u>second best</u> results are underlined. In the respective paradigms, the ***SOTA models*** are marked in italics.

Models	BART-base		BART-large	
	RAMS	WikiEvents	RAMS	WikiEvents
PAIE (Ma et al., 2022)	<u>90.5</u>	92	91.9	92.6
BART-Gen (Li et al., 2021)	88.2	87.8	92	**93.4**
Fusion Generatively Biased	90.2	<u>92.7</u>	92.2	<u>93.3</u>
Fusion Balanced	**90.7**	**93.6**	**92.8**	92.6
Fusion Selectively Biased	<u>90.5</u>	92.1	<u>92.4</u>	92.1

Table 2: Comparison of the Ratio (Arg-C/Arg-I) Across Models on RAMS and WIKIEVENTS.

Models	BART-base	BART-large
PAIE (Ma et al., 2022)	7340	17000
BART-Gen (Li et al., 2021)	4453	10021
Fusion	5352	12518

Table 3: Comparison of GPU Memory Usage (MB) across different models.

(2) Compared to generative models, our fusion model *effectively guides outcome generation through the selection part, significantly boosting extraction performance*. As illustrated in the Table 1, our model outperforms generative counterparts across various metrics. Notably, when pitted against the SOTA generative model Retrieval-augmented, our model attains an improvement of 3.3%~4.2% and 1%~2%, reinforcing the notion that *the integration of selective methods can lead to more accurate and precise outcomes*.

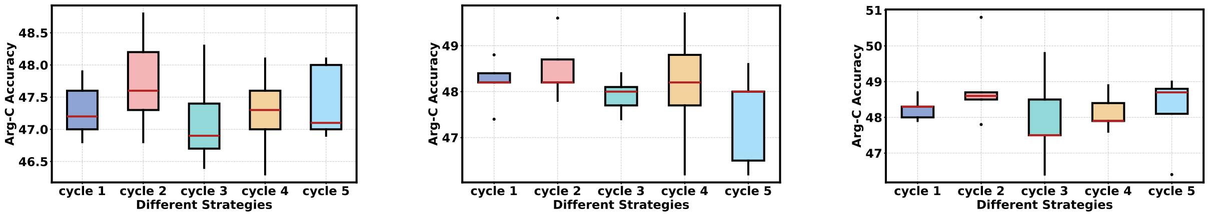
Selection	Generation	Fusion sel	Fusion msk	Arg-I	Arg-C
✓	✓	✓	✓	**53.6**	**48.6**
✓	✓	✓		52.0	47.3
✓	✓		✓	24.8	22.8
✓		✓		50.7	45.7
	✓		✓	29.7	26.3

Table 4: Ablation studies are conducted on the RAMS dataset using the Fusion Balanced model based on BART-base.

3.3 Analysis

Ablation Study We perform ablation studies on key components of the model, including the Selection Part, Generation Part, and the selection logits and mask logits in the Fusion Learning. As shown in Table 4, the full configuration, which includes all components, achieves the best results across all evaluation metrics.

The experiments reveal that the selection part establishes a strong performance baseline. However, incorporating the generation part improves the model's ability to capture complex semantic relationships, resulting in overall better performance. This demonstrates that combining the two approaches harnesses their respective strengths: the precision of the selection part and the semantic richness of the generation part, ultimately leading to a more robust and adaptable model.



(a) Learning rate of 5e-5 (b) Learning rate of 3e-5 (c) Learning rate of 2e-5

Figure 4: Performance of different fusion strategies on RAMS

Fusion Strategy Our approach to constructing the fusion model involves a progressively unified learning strategy, where we discover that the dynamic nature of the loss function necessitates a similarly adaptive learning rate strategy. The loss function is not static but evolves with training iterations, shifting the model's convergence point. In this context, we employ a Cosine with Hard Restarts (Gotmare et al., 2019) learning rate scheduling strategy, assessing its impact on the fusion model's performance by varying the cycle lengths. Experiments on the RAMS with different learning rate strategy cycles on the BART-base model reveal significant trends. As depicted in Figure 4, longer cycles lead to more dispersed performance distributions, suggesting that increased cycle lengths are not always beneficial at these settings. Particularly, under cycle 2 settings, the model not only shows higher stability but also reaches a relatively higher performance mean. In contrast, the single-cycle learning strategy (cycle 1) performs worse in accuracy compared to cycle 2, indicating that traditional single-cycle learning rate adjustments may not be suitable for fusion models. A more adaptive, multi-cycle learning rate strategy could be crucial for optimizing performance in such models.

4 Related Work

Event Argument Extraction Event Argument Extraction (EAE) focuses on identifying and extracting arguments from texts, related to specific events (Zheng et al., 2019). EAE operates under four primary paradigms: (1) **Sequence Labeling** (Wang et al., 2020; Shi and Lin, 2019), which annotates event-related arguments in texts, marking relevant segments; (2) **Token Classification** (Lin et al., 2020; Xu et al., 2021; Ding et al., 2023; Yang et al., 2021), categorizing each word by argument type for targeted extraction; (3) **Machine Reading Comprehension (MRC)** (Du and Cardie, 2020; Liu et al., 2020; Wei et al., 2021; Liu et al., 2021; Ma et al., 2022), formulating questions related to the event to extract specific text spans as answers; and (4) **Sequence to Sequence** (Lu et al., 2021; Paolini et al., 2021; Li et al., 2021), a newer approach that treats EAE as a sequence generation task, focusing on serializing text outputs to identify precise event-related information. Each paradigm offers distinct methods for dissecting and understanding event-themed texts.

Hybrid Model Pointer-Generator Networks (See et al., 2017) effectively bridge the gap between extractive and abstractive text summarization methods (Qiu and Yang, 2022). Extractive summarization involves selecting significant sentences, while abstractive summarization focuses on generating concise, coherent summaries. The Pointer-Generator Network model, building on pointer networks (Vinyals et al., 2015), innovatively addresses challenges in both approaches. It combines direct copying from source texts to enhance accuracy and manage out-of-vocabulary words with the generation of new content. Our model is inspired by this approach. However, we integrate the two methods differently by leveraging pre-trained models to further enhance their combination.

5 Conclusion

In conclusion, our research presents a Fusion Selection-Generation-Based Approach for Event Argument Extraction, merging selective and generative methods. Empirical evaluations on the RAMS and WIKIEVENTS indicate improved performance and efficiency. This study contributes to the EAE field by demonstrating the practicality of integrating different approaches. In future work, we plan to design a more suitable fusion method and adapt our fusion model to other domains, thereby exploring broader applications and achieving deeper integrations in information extraction.

Limitations

Firstly, the generation part treats event arguments of varying lengths as a single mask, leading to substantial information loss. While our fusion approach has shown benefits, there remains a need for a more effective method to minimize this loss of information.

Second, our experiments reveal that different datasets exhibit varying biases towards selection and generation parts. This implies a significant reliance on adjusting the fusion parameter λ, requiring multiple modifications to optimize performance for different datasets. Such dependency indicates the need for a more adaptive approach in balancing the strengths of both selection and generation parts across diverse data contexts.

Furthermore, extracting multiple arguments for the same role in complex sentences remains a challenge. Although the number of extracted arguments can be increased by modifying the reserved argument slots in the prompt, a more flexible approach is still needed to address this issue effectively.

References

Guoxuan Ding, Xiaobo Guo, Gaode Chen, Lei Wang, and Daren Zha. 2023. HAEE: Low-resource event detection with hierarchy-aware event graph embeddings. In *International Semantic Web Conference*, pages 61–79. Springer.

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.

Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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.

Qian Li, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng, Shu Guo, Lihong Wang, Amin Beheshti, et al. 2022. A survey on deep learning event extraction: Approaches and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

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.

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

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

Jian Liu, Yufeng Chen, and Jinan Xu. 2021. Machine reading comprehension as data augmentation: A case study on implicit event argument extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2716–2725, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

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.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.

Dong Qiu and Bing Yang. 2022. Text summarization based on multi-head self-attention mechanism and pointer network. *Complex & Intelligent Systems*, pages 1–13.

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

Yubing Ren, Yanan Cao, Ping Guo, Fang Fang, Wei Ma, and Zheng Lin. 2023. Retrieve-and-sample: Document-level event argument extraction via hybrid retrieval augmentation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–306, Toronto, Canada. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *ArXiv preprint*, abs/1904.05255.

Tian-Xiang Sun, Xiang-Yang Liu, Xi-Peng Qiu, and Xuan-Jing Huang. 2022. Paradigm shift in natural language processing. *Machine Intelligence Research*, 19(3):169–183.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2692–2700.

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

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, Online. Association for Computational Linguistics.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 2021. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. 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 3533–3546, Online. Association for Computational Linguistics.

Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. 2021. Document-level event extraction via parallel prediction networks. 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 6298–6308, Online. Association for Computational Linguistics.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019a. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, Florence, Italy. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2EDAG: An end-to-end document-level framework for Chinese financial event 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 337–346, Hong Kong, China. Association for Computational Linguistics.

Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. 2020. Temporal common sense acquisition with minimal supervision. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7579–7589, Online. Association for Computational Linguistics.

A Dataset and Model

A.1 Dataset Statistics

In this study, two significant datasets are utilized for document-level event argument extraction. RAMS includes 9,124 news examples, covering 139 event types and 65 argument roles, offering a broad perspective on real-world events. WIKIEVENTS is compiled from 246 English Wikipedia articles and features 50 event types and 59 argument roles. This dataset provides a unique view into encyclopedic events. Both datasets are essential for understanding the complexity and diversity of event argument extraction in different contexts. Table 5 shows their detailed statistics.

Dataset	#Doc	#Event	#Argument	Split
RAMS	3,194	7,329	17,026	Train
	399	924	2,188	Dev
	400	871	2,023	Test
WikiEvents	206	3,241	4,542	Train
	20	345	428	Dev
	20	365	566	Test

Table 5: Statistics of RAMS and WIKIEVENTS datasets.

A.2 Details of Baseline Models

We compare our model with following previous models. (1) BERT-CRF (Shi and Lin, 2019): This model employs a BERT-based model employing BIO-styled sequence labeling for multi-label classification. The model's architecture synergizes the robust contextual embeddings of BERT with the sequence decoding capabilities of a Conditional Random Field (CRF), aiming to enhance the precision of classification. (2) EEQA (Du and Cardie, 2020): Pioneering the application of Question Answering (QA) mechanisms to the sentence-level EAE task, EEQA diverges from traditional classification-based approaches. By reframing EAE as a QA problem, it seeks to capitalize on the innate capability of QA systems to discern fine-grained information within a text. (3) PAIE (Ma et al., 2022): Extending from EEQA, this model introduces a prompt tuning strategy specifically for EAE. It reimagines the multi-label classification challenge by embedding prompts that guide the model to generate more contextually relevant and precise arguments. (4) BART-Gen (Li et al., 2021): This model approaches EAE through a sequence-to-sequence lens, utilizing the BART-large framework. The objective is to produce arguments that not only align with the predefined format but also encapsulate the nuances of the events being modeled. The BART-Gen demonstrates a significant stride in generating coherent and contextually accurate arguments. (5) Retrieval-augmented (Ren et al., 2023): A novel adaptive hybrid retrieval augmentation paradigm that adaptively samples pseudo demonstrations from continuous space for each training instance to improve the analogical capability of the model.

A.3 Implementation Details

Hyperparameter	**Value**
Batch size	4
Weight decay	0.01
Training steps	10,000
Optimizer	AdamW
Scheduler	Cosine with Hard Restarts
Warmup steps	0.1
Number of cycles	2
Max span length	10
Max gradient norm	5.0
Max encoder seq length	500
Max decoder seq length	100

Table 6: Hyperparameters used in the experiments.