Document-Level Event Extraction Based on Multi-instance Learning

Abstract

Current research focuses on utilizing intra-document information for event extraction, but it has limitations in capturing the complexity and diversity of events because it overlooks the relationships between documents. Additionally, current instance learning methods for event extraction primarily focus on similarity-based instance retrieval, failing to emphasize comprehensive model learning, and a single measure of similarity cannot fully reflect the semantics of a document. To address these issues, this paper proposes an event extraction model based on multi-instance learning, exploring the connections between documents through event types and event arguments. We designed multiple instance selection strategies and construction methods to enable the model to achieve a more thorough understanding of events. Furthermore, we implemented a two-stage training approach to optimize the model's ability to learn from instances obtained through different instances. Experiments conducted on the RAMS and WIKIEVENTS datasets demonstrate that our method surpasses the current state-of-the-art models in terms of F1 scores, validating its effectiveness and superiority.The source code is available on GitHub.[1]

1 Introduction

Early work on document-level event extraction(DEE) focused on modeling each event independently(Du et al., 2021; Li et al., 2021), neglecting the global context. Recent studies have primarily concentrated on the Cross-Sentence Argument Role problem(Xu et al., 2022), exploring how to effectively utilize intra-document information (Liu et al., 2023; Huang et al., 2023). However, these methods often overlook the intrinsic connections between documents, failing to fully exploit the potential information among the context.

There exists a significant amount of interrelated information among document data. For example, in Figure 1, "President Kennedy was assassinated in Texas" is categorized as a death caused by a violent event. For other documents mentioning the same event type, such as "Humayun Khan" due to a "suicide bomber," different causes of death are proposed. Furthermore, same category events like "Timoshenkov died 24 hours after being rescued" provide supplementary information. the model can learn multiple causes leading to death events, helping it to comprehensively understand events and enhance the precision of semantic capture.

Event type: Life.Die.deathcausedbyviolentevents

Main Sentence
[...] It 's not clear if President Kennedy ever saw the message.Two weeks later, he would be assassinated in Texas. [...]
Same Type sentence
Ghazala Khan , whose son , U.S. Army Capt . Humayun Khan, was killed by a suicide bomber in Iraq in 2004. [...]

Event type: Life.die.nonviolentdeath

Same Category Sentence
[...] The doctors at Russian Khmeimim airbase in Latakia province fought for Timoshenkov 's life for over 24 hours, but he passed away on June 16, it added. [...]

Figure 1: Examples of multi-level knowledge in cross-document contexts for understanding death events.

Related studies indicate that multi-instance learning can enable models to capture the connections between data. (Su et al., 2022) significantly enhanced model performance by retrieving similar instances and performing label interpolation. (Chen et al., 2022) improved model performance and generalization in low-resource settings by constructing knowledge bases and retrieval mechanisms.

Although multi-instance learning has been proven effective, it has not yet shown significant results in document-level event extraction. (Zhao

[1] https://github.com/shdmm00/MILE

et al., 2023) selected a demonstration based on context and argument roles in low-resource tasks, but a single instance is prone to noise. (Du et al., 2022) filtered templates related to the input document as concatenated inputs. (Ren et al., 2023) used similarity retrieval to find the most relevant samples. These approaches have limitations in capturing the complexity and diversity of events, as a single measure of similarity cannot comprehensively reflect the semantics of a document, and document similarity does not equate to event similarity. Furthermore, effectively constructing and thoroughly learning from multiple instances is more crucial than simply retrieving instances.

To enhance the interconnections among document data, this study proposes a **M**ulti-**I**nstance **L**earning **E**vent **E**xtraction model. Specifically, after document encoding, instances are selected from the instance vector database using various strategies. Each instance interacts with the main data through a cross-attention mechanism and is scored by an instance selector. Ultimately, the selected instances are concatenated with the main data. Using different instance selection methods, a two-stage training process was implemented to leverage the information from the instances fully. The main contributions of this paper are as follows:

1) We introduced Multi-Instance Learning to the task of document-level event extraction and explored various instance construction and selection methods, including the development of an instance selector to optimize instance selection.

2) We implemented a two-stage training approach, enabling the model to adapt and fully utilize the multidimensional information obtained through different instance selection methods.

3) Experimental results on multiple chapter-level datasets validated the effectiveness of this method, with F1 scores surpassing current state-of-the-art approaches.

2 Related work

Document-Level Event Extraction The primary models for event extraction include discriminative and generative models. Discriminative models use event triggers for sequence labeling (Du and Cardie, 2020; Veyseh et al., 2021) or span-based prediction (Zhang et al., 2020; Ebner et al., 2020; Liu et al., 2023). (Zheng et al., 2019) introduced a transformer-based architecture for serial prediction, while (Wei et al., 2021) redefined the task as comprehension. (Ren et al., 2022) integrated argument roles into encoding. (Xu et al., 2022) and (Yang et al., 2023) combined AMR graphs to address long-distance dependencies.

Generative models excel in event extraction by producing richer outputs and capturing complex relationships through iterative or parallel generation. (Yang et al., 2021) proposed an encoder-decoder framework for parallel extraction, while (Ma et al., 2022) and (Zeng et al., 2022), along with (Li et al., 2021) and (Du et al., 2021), investigated the use of prompts for generation and framed the problem as conditional generation.(Huang et al., 2023) addressed training inadequacies with a pre-filling strategy.

Demonstration-based Learning. Inspired by GPT-3's contextual capabilities (Brown et al., 2020), In-Context Learning enables models to learn from few-shot demonstrations.(Lester et al., 2021) explored instance prompt design, while (Zhao et al., 2021; Lu et al., 2022) examined its impact on the model. (Lee et al., 2022; Zhang et al., 2022) explored using demonstrations for NER tagging. (Su et al., 2022; Li et al., 2023) employed KNN for instance retrieval, and (Chen et al., 2022) created a prompt-based knowledge base. Combining the advantages of generative models and multi-instance learning, we propose our model.

3 Methodology

Given a document, the objective of event argument extraction is to: 1) Identify arguments and their roles{ arg1, arg2, ... } within an event. 2) Classify these arguments to populate predefined templates for each event type $\mathcal{E}$.

3.1 Model Architecture

We propose an end-to-end event extraction model that leverages the capabilities of generative models and incorporates instance-based learning methods. Following the ontology by (Li et al., 2021), we adopt argument templates defined therein. These templates include numerical labels and slot mappings for each argument, which correspond to specific argument roles. As shown in Figure 2, each input document C_i is concatenated with a template T_i. This combined input is then processed by the encoder to obtain the vector representation $\mathbf{H}_i$ as follows:

$$\mathbf{H}_i = \text{Encoder}([T_i; C_i]) \quad (1)$$

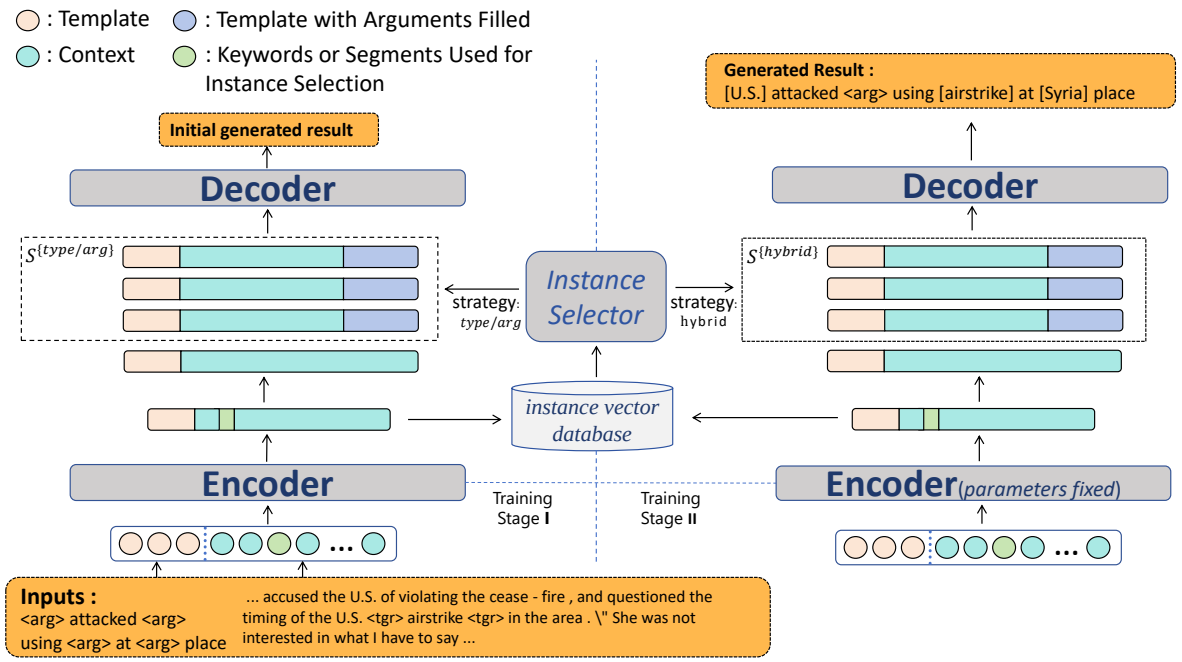

Figure 2: Illustration of the MILE. The input document, concatenated with event templates, is fed into the encoder to obtain embeddings. Relevant instances are selected from a pre-constructed database and concatenated with the embeddings for decoding. MILE uses a two-stage training approach: first, simple strategies for instance selection. In the second stage, the encoder is fixed, and complex selection methods are employed.

Subsequently, from a pre-constructed instance vector database, relevant instance vectors $\mathcal{S}_i$ are selected based on different selection strategies. These selected instance vectors are then concatenated with $\mathbf{H}_i$ and fed into the decoder to generate the target event sequence Y_i:

$$y_{ij}, h_{ij} = \text{Decoder}([\mathbf{H}_i; \mathcal{S}_i; h_{i,j-1}]) \quad (2)$$

y_{ij} is the j-th token generated, and h_{ij} is the hidden state at the j-th step of the decoder. When constructing the input, we use <trg> to mark the event trigger and <arg> to mark the arguments to be filled.

3.2 Instance Vector Database Construction

We utilized FAISS(Johnson et al., 2019) to construct an instance vector database. Each database instance includes structured templates for the context above and below, encoded with a fixed pre-trained encoder to maintain data independence and prevent learning correlations. Each instance $\mathbf{s}$ is represented as:

$$\mathbf{s} = \langle\text{instance}\rangle\text{concat}(T; C; T') \quad (3)$$

T' is the template for the context after padding. Each instance is prepended with the $\langle$instance$\rangle$ tag. We explored various retrieval strategies and ultimately opted for the HNSW to ensure the efficiency and accuracy of the retrieval process.

The database construction employs global data utilization for diverse coverage and relevant sentence selection focusing on trigger words to enhance precision. We also investigated the impact of different instance construction methods, including context-only and both template and context, on model performance.

3.3 Instance Selection Strategies

Pre-selection Initially, we perform instance pre-selection using embedded vectors of keywords or segments, utilizing vector similarity to capture deeper semantic relationships in texts. When performing instance selection, we employ random selection strategies to increase the diversity of retrieved instances and iterate through pre-selected vector instance set. This approach ensures that the model learns from a wide range of instances.

Similarity-Based Selection Our model employs the FAISS database for efficient instance selection based on similarity search, serving as a benchmark for comparison in similarity selection methods.

Type-Based and Category-Based Selection Document contexts contain multi-level knowledge,

and the type of event determines its semantic direction. To ensure that the retrieved instances are highly relevant to the input context, we select instances with the same event type from the instance database for target event types. Additionally, We introduce category-based instance selection strategies to select instances related to the same event category, enhancing the model's ability to distinguish between event types and improving its generalization performance.

Argument Keyword-Based Selection For this selection approach, we first analyze each instance in the database using various prediction methods to identify potential argument keywords, then aggregate these keywords through a weighted sum to form a comprehensive set of argument keywords. This set combines results from multiple methods, selecting instances with the most relevant argument keywords to ensure high semantic relevance to the target event. We calculate the confidence score of each argument keyword based on its consistency and frequency, and select instances containing keywords with high confidence scores. The process is represented by:

$$\mathcal{S} = i \mid \sum_{m=1}^{M} \alpha_m \cdot \text{ind}(k \in \mathbf{K}^m) > \theta,\ k \in \mathbf{K}_i \quad (4)$$

where α_m is the weight of the m-th prediction method, M is the number of methods, and $\text{ind}(k \in \mathbf{K}^m)$ is 1 if the keyword k is predicted by the m-th method, and 0 otherwise. This approach ensures that the argument keywords are evaluated based on their importance and relevance across different prediction methods, streamlining the selection of instances highly relevant to the target event.

Hybrid Selection In addition to type-based and argument keyword-based instance selection strategies, we propose a hybrid selection strategy. This strategy combines the advantages of both individual strategies, enabling the model to select instances from multiple perspectives, thereby improving the accuracy and relevance of event extraction.

3.4 Instance Scoring Selector

We designed an instance scoring selector that uses a cross-attention mechanism to select the most optimal instances during prediction, thereby providing the model with the most relevant and useful information. First, we calculate the attention score between the hidden state $\mathbf{H}_i$ of each instance selected and the input instance. Next, we process these attention scores through average pooling and max pooling to obtain a score for each instance, denoted as γ:

$$\gamma = \text{Pool}_{AM}\left(\text{softmax}\left(\frac{\mathbf{H}_i \cdot \mathcal{S}^\top}{\sqrt{d_{\text{key}}}} \cdot \mathbf{T}\right)\right) \quad (5)$$

where Pool_{AM} is a combined pooling operation that integrates both average and max pooling. d_{key} is the dimensionality of the $\mathbf{H}_i$. Respectively, and $\mathbf{T}$ denotes the temperature factor.

This design ensures the model considers both the most relevant instances and overall relevance, with the cross-attention mechanism better-capturing instance relevance and usefulness compared to traditional methods.

3.5 Two-Phase Training Approach

To enhance the model's learning effectiveness and adaptability to various selected instances, we adopt a two-phase training strategy.

In the first phase, we concentrate on training the model with straightforward instance selection methods. The goal is to enable the model to recognize and process the basic structure and semantics of instances while adapting to their diversity and complexity. In this phase, we optimize the following negative log-likelihood loss function:

$$\mathcal{L} = -\sum_{i=1}^{N} \log p\left(Y_i \mid X_i, \mathcal{S}_i^{\{\text{type/arg}\}}, \theta\right) \quad (6)$$

where Y_i is the label of the i-th sample, and X_i is the input document. θ represents the model parameters, and $\mathbf{S}_i^{\{\text{type/arg}\}}$ denotes the instances selected using type-based or argument keyword-based instance selection methods.

In the second phase, we fix the encoder parameters and continue to optimize the same loss function, while further training the model using the complex instance selection set $\mathcal{S}_i^{\{\text{hybrid}\}}$. This phase aims to enable the model to learn deeper semantic relationships and complex event structures.

4 Experiment

4.1 Experimental Setup

Dataset and Evaluation Metrics We used two datasets: **RAMS** (Ebner et al., 2020) with cross-sentence arguments and one event per document,

and **WIKIEVENTS** (Li et al., 2021) with multiple events annotated using the *DARPA KAIROS* ontology.

For evaluation, we followed established criteria. Span F1 requires exact span matches, potentially misclassifying correct predictions. Thus, we used Head F1 (based on the span's head word) and Coref F1 (crediting coreferential spans with gold-standard arguments). For Wikievents, we report Head F1 and Coref F1 for argument identification (Arg IF) and classification (Arg CF).

Baselines We compare **MILE** several models in three categories: (1) QA-based model: **FEAE**(Wei et al., 2021), **BERT-QA**(Du and Cardie, 2020) (2) Discriminative Model: **BERT-CRF**(Shi and Lin, 2019), **Two-Step**(Zhang et al., 2020), **BERT-CRF**$_{\text{TCD}}$ and **Two-Step**$_{\text{TCD}}$(Ebner et al., 2020), **TSAR**(Xu et al., 2022), **SCPRG**(Liu et al., 2023), **TARA**(Yang et al., 2023) (3) Generation model: **BART-Gen**(Li et al., 2021), **(Chen et al., 2022)**, **EA^{2}E**(Zeng et al., 2022)

4.2 Main Result

We conducted the main experiments on the RAMS and Wikievents datasets, aiming to evaluate the model's performance in argument identification and classification tasks across different datasets. Table 1 summarizes the experimental results on the RAMS dataset, where **MILE** demonstrates superior performance compared to previous generative models such as **BART-Gen** and **TSAR**, particularly excelling in the Head F1. On the test set, **MILE** achieved Span and Head F1 scores of 51.37 and 60.26 respectively, outperforming the current leading model. A high Head F1 score indicates that **MILE** can accurately recognize and classify the most critical elements in events, demonstrating the model's powerful comprehension ability. Furthermore, **MILE**'s outstanding performance in the Head F1 score on the RAM dataset highlights its capability in accurately identifying the core words of event arguments.

Similarly, on the Wikievents dataset as detailed in Table 2, **MILE** achieved 76.79 in argument identification (Arg IF), surpassing most comparative models. The Coref F1 evaluation metric on the Wikievents dataset adds an assessment of coreferential mentions, and a higher Coref F1 score indicates that the model has a better capability in understanding and linking the same or related entities in the text.

Method	**Dev**		**Test**	
	Span F1	Head F1	Span F1	Head F1
BERT-CRF	38.1	45.7	39.3	47.1
BERT-CRF-TCD	39.2	46.7	40.5	48.0
Two-Step	38.9	46.4	40.1	47.7
Two-Step-TCD	40.3	48.0	41.8	49.7
FEAE	-	-	47.40	-
TSAR	49.23	56.76	51.18	58.53
SCPRG	50.53	57.66	52.32	59.66
BART-Gen	-	-	48.64	57.32
MILE (ours)	49.51	**57.89**	51.37	**60.26**

Table 1: Main results of RAMS.

Method	**Arg IF**		**Arg CF**	
	Head F1	Coref F1	Head F1	Coref F1
BERT-CRF	69.83	72.24	54.48	56.72
BERT-QA	61.05	64.59	56.16	59.36
BERT-QA-Doc	39.15	51.25	34.77	45.96
TSAR	76.62	75.52	69.70	68.79
SCPRG	77.26	76.10	70.92	70.08
TARA	78.64	76.71	73.33	71.55
BART-Gen	71.75	72.29	64.57	65.11
EA^{2}E	74.62	75.77	68.61	69.70
(Chen et al., 2022)	-	-	68.04	-
MILE (ours)	76.31	**76.79**	70.53	71.33

Table 2: Main results of Wikievents.

As shown in the table, **MILE** has achieved significant improvements compared to previous generative models and multi-instance learning models. Thus, methods based on multi-instance learning help the model to better understand and handle semantic complexity, maintaining the consistency and accuracy of information.

The underperformance of our model in some evaluations compared to **TARA** can be explained: 1)**TARA** is a discriminative model, which is usually directly optimized for the prediction task and may be more precise in specific tasks. 2) **MILE** achieved comparable Coref scores to **TARA**, indicating similar event understanding capabilities between the two models, with only a deficiency in boundary identification. 3) **MILE** focuses on cross-document clues and does not consider long-distance dependencies within a document as **TARA** does. However, as a generative model, it has the potential to more effectively leverage the capabilities of large-scale pre-trained models.

Selection Method	N = 1		N = 2		N = 3		N = 4		N = 5		N = 6	
	SP1	HF1	SP1	HF1	SP1	HF1	SP1	HF1	SP1	HF1	SP1	HF1
Similarity	48.58	57.10	48.69	57.33	48.87	57.56	48.34	57.30	48.25	57.04	47.89	56.81
Type-Based	48.69	57.66	49.28	58.06	49.97	58.42	**50.51**	58.47	49.34	58.78	50.36	59.64
Keyword-Based	48.60	57.44	49.35	58.49	50.11	59.17	50.46	59.63	50.30	**59.74**	49.70	58.40
Category-Based	48.92	57.65	49.73	58.98	50.37	59.40	50.28	59.31	50.15	59.26	50.02	59.39
Hybrid	48.61	57.36	49.11	58.24	49.43	58.55	49.72	58.47	49.41	58.44	50.07	58.89

Table 3: Performance comparison across different selection methods and instance numbers.

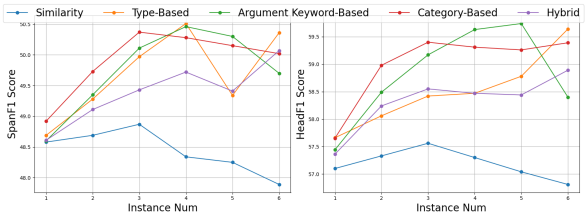

Figure 3: Instance selection performance trends.

Instance-Dim	$T+C+T'$		$C+T'$		C	
	HF1	CF1	HF1	CF1	HF1	CF1
48*	65.81	65.49	65.81	66.00	64.87	65.02
96	69.86	70.55	**70.53**	**71.33**	70.18	70.96
128*	66.16	66.12	65.39	65.12	65.73	65.80
256	68.40	68.71	68.74	69.05	68.20	68.59

Table 4: The impact of Instance Dimension and Construction. * indicates pooling operation.

4.3 Discussion on Instance Integration

In this section, we analyzed the impact of instance features on model performance. This revealed the strengths and limitations of each strategy, as well as the effects of instance dimensions and construction methods on model efficacy.

Analysis of Selection and Quantity We thoroughly explored the impact of various instance selection methods on model performance. Figure 3 illustrates the performance trends of five instance selection methods under different numbers of instances. It can be observed from the figure that *Type-Based* and *Keyword-Based* selections generally perform well in most cases, while *Similarity* selection perform poorly.

Table 3 shows the performance comparison across different selection methods and instance numbers. In Table 3, the Span F1 and Head F1 scores for the *Type-based* selection increased by 1.28 and 0.76 when the number of instances increased from 1 to 3. Increasing from 1 to 3 instances improved the performance of most selection methods, indicating that a moderate increase helps the model better understand and extract event information. However, increasing to 5 or 6 instances, such as the Span F1 and Head F1 scores of 50.36 and 59.64 for *Type-based* showed a flattening growth trend. This suggests that too many instances may introduce noise or cause overfitting, indicating the need for a balance between instance number and performance.

In selection methods comparative analysis, the *Type-based* method performed best with 6 instances, achieving Span F1 and Head F1 scores of 50.36 and 59.64. This indicates that more instances help capture event diversity, enhancing the model's generalization capability. Respectively, the *Keyword-Based* achieved peak performance with four instances, suggesting that an appropriate number of instances aids the model in capturing key event arguments. The *Category-Based* performed best with 3 instances, with Span F1 and Head F1 scores of 50.37 and 59.40, respectively, possibly due to incorporating diversity while maintaining event relevance. The *Hybrid* method performed best with 6 instances, combining event types and arguments to provide multi-dimensional information, maintaining high performance in most scenarios. Results using *Similarity* showed no significant improvement and even declined, confirming that relying solely on document similarity does not enhance model performance. These findings reveal the strengths and limitations of different selection strategies, providing valuable guidance for the design of more efficient models.

Dimensions and Construction This section investigates the impact of instance dimensions and construction on model performance. Instance dimensions capture information at different document levels. *Dim 96* uses only the event sentence,

capturing directly related information, while *Dim 256* uses the entire document, providing broader context but potentially including irrelevant information. Other dimensions, such as *Dim 48* and *Dim 128*, are obtained through pooling of the aforementioned approaches, aiming to explore how different information densities impact model performance.

The experimental results in Table 4 show that when the instance dimension is *Dim 96*, the model achieved the highest Head F1 score, which is 70.53, indicating that focusing directly on the sentence where events occur can minimize noise interference and improve the model's precision. With *Dim 256*, the model's Head F1 score decreased to 68.74, possibly because of excessive irrelevant information. Intermediate dimensions obtained through pooling (e.g., *Dim 128*) generally perform worse than *Dim 96* and *Dim 256*, with a Head F1 score of 66.16. This decline may be due to key information loss during pooling or inconsistent information hierarchy, making it difficult for the model to utilize effectively.

This section explores the impact of instance construction methods, comparing the effects of complete construction $T+C+T'$ with those containing only contexts C or event contexts with output templates $C+T'$. The experimental results show that $C+T'$ achieved the highest Coref F1 at 71.33, followed by C alone at 70.96, and $T+C+T'$ at 70.55. This suggests that $C+T'$ provides additional semantic cues, enhancing the model's understanding of events. Moreover, the inclusion of empty templates T may have confused the decoder, resulting in less favorable outcomes. These results highlight the importance of selecting relevant instances, as the right instance dimension and quantity significantly influence model performance.

4.4 Analysis of Two-Stage Training

This section of the research investigates the application and effectiveness of a two-stage training method across various instance selection strategies, followed by a detailed analysis of the experimental results in Table 5.

Among the findings, the *Category-Based + Hybrid* combination demonstrated the best performance with a Head F1 score of 76.31 and a Coref F1 score of 76.79, indicating its superiority in event head identification and coreference accuracy. This may be attributed to the strategy's high efficiency in handling the relevance and complexity of events.

selection Method	Arg IF		Arg CF	
	HeadF1	CorefF1	HeadF1	CorefF1
Similarity				
+ Type-Based	73.59	73.87	67.80	68.23
+ Category-Based	73.62	73.91	67.88	68.34
+ Hybrid	72.48	72.75	66.92	67.29
Type-Based				
+ Category-Based	76.02	75.18	69.75	69.91
+ Hybrid	**76.31**	75.13	69.99	70.15
Category-Based				
+ Type-Based	75.08	75.24	69.12	69.28
+ Hybrid	75.88	**76.79**	70.48	**71.33**
Keyword-Based				
+ Type-Based	74.56	74.72	69.07	69.23
+ Category-Based	74.64	74.80	69.16	69.32
+ Hybrid	75.09	75.25	**70.51**	70.77

Table 5: Impact of two-stage training on performance.

It considers the direct type of the event and includes related subcategories, providing richer contextual information for better event understanding and description generation. By combining event categories with a hybrid selection strategy, it comprehensively covers various situations and variants related to the target event. In contrast, the *Similarity+Hybrid* combination performed less effectively, with Head F1 and Coref F1 scores of 72.48 and 72.75. This strategy may have struggled to distinguish key information from noise due to the limitations of similarity searches and the complexity of the Hybrid approach, leading to decreased performance. Additionally, while the *Category-Based + Hybrid* optimized relevance and quality through refined instance selection, the *Similarity + Hybrid* may have introduced too much irrelevant information, affecting the model's accuracy.

Other combinations, such as *Type-Based + Category-Based* and *Keyword-Based + Hybrid*, also performed well, with Head F1 scores of 75.02 and 75.09. This suggests that considering both type and keyword information helps the model capture core elements and context more accurately. The *Argument Keyword Based + Hybrid* strategy, in particular, enhances the model's ability to identify event arguments by optimizing for important event attributes.

The two-stage training method significantly influenced the model's performance. The first stage focused on identifying basic event structures, while the second stage optimized the model with a hybrid strategy to handle complex event relationships and deeper semantic information.

Method	Arg IF		Arg CF	
	Head F1	Coref F1	Head F1	Coref F1
MILE	76.31	76.79	70.53	71.33
w/o ISS	76.43	76.21	70.32	71.22
w/o TTA	75.09	75.47	69.56	70.25
w/o MIS	71.95	72.03	65.80	66.91

Table 6: Ablation Study on WikiEvents.

Method	Dev		Test	
	Span F1	Head F1	Span F1	Head F1
MILE	49.51	57.89	51.37	60.26
w/o ISS	49.23	57.69	50.82	59.97
w/o TTA	48.10	56.22	50.51	59.63
w/o MIS	46.67	54.81	48.20	57.03

Table 7: Ablation Study on RAMS.

4.5 Ablation Experiments

In this section of the study, we explored the significance of model components through ablation experiments conducted on the RAMS and Wikievents datasets. The results in Table 6 showed that removing the Instance Scoring Selector (ISS) and Two-stage Training Approaches (TTA) led to slight F1 score drops of 0.21 and 0.11 in Head F1 and Coref F1 on WikiEvents. This suggests ISS contributes to performance but isn't crucial. Removing TTA caused larger drops of 0.97 and 1.08 in Head F1 and Coref F1. The most significant drop 4.42 in Coref F1, occurred after removing the Multi-Instance learning Strategy MIS, highlighting its key function in understanding complex semantics.

The ablation experiments on the RAMS dataset revealed similar trends. Removing the ISS decreased Span F1 and Head F1 by 0.55 and 0.29. Removing TTA led to larger declines of 0.86 and 1.63. The most significant drops, 2.17 and 3.23, occurred after removing MIS, confirming its importance in complex semantic understanding. These results show that MIS are crucial for enhancing model performance across different datasets and tasks. This suggests that the multi-instance learning approach enhances event extraction by directing the decoder to focus on key segments and better capture critical elements.

4.6 Distributions in MILE

To validate the efficacy of our model, we compared the attention maps generated by our model and the base model. The results in Figure 4 indicated that our model exhibits a more concentrated distribution of attention on tokens related to the event. Specifically, the attention map on the left (**Ours**) demonstrates higher attention weights concentrated at particular positions, which correspond to tokens associated with the event, whereas the base model on the right (**Base**), which is not trained using the multi-instance strategy, displays a more scattered and unfocused distribution of attention.

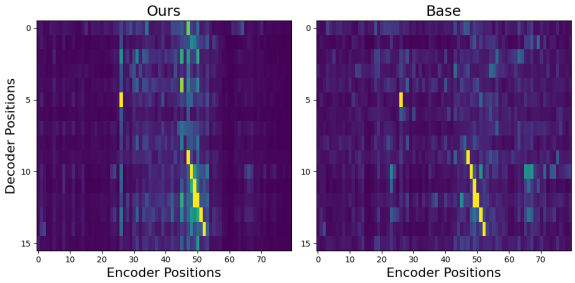

Figure 4: Comparison of attention maps between MILE and the base model.

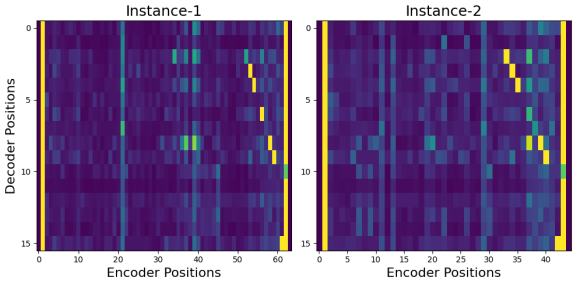

Figure 5: Attention distribution of MILE on two *Type-Based* instances.

We plotted the attention distribution of the multi-instance learning model on two *Type-Based* instances in Figure 5. The attention maps for **Instance-1** and **Instance-2** show a similar pattern, indicating that consistent attention distribution across same-type instances helps the model capture critical event information more accurately.

5 Conclusion

This paper proposes a novel multi-instance learning method for document-level event extraction. By encoding event documents and selecting diverse instances, this method explores semantic relationships and enhances model learning. We validated the effectiveness of different instance selection methods. Our model, MILE, outperforms existing methods on multiple datasets, excelling in argument identification and classification.

Limitations

Although our method is effective for document-level event extraction tasks, it has some limitations. Using an instance vector database requires significant storage space, especially when handling large datasets. Preprocessing steps such as instance selection and vector generation are necessary for different datasets, which can be cumbersome and time-consuming, increasing the complexity of model deployment and potentially limiting its application in resource-constrained environments. Improper instance selection may introduce noise, reducing the model's effectiveness. Future work will explore autonomous instance selection, optimize selection algorithms, and reduce dependence on large-scale databases. We also plan to extend this method to other information extraction tasks, such as relation extraction and multilingual extraction.

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.

Xiang Chen, Lei Li, Ningyu Zhang, Xiaozhuan Liang, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Decoupling knowledge from memorization: Retrieval-augmented prompt learning. *Advances in Neural Information Processing Systems*, 35:23908–23922.

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.

Xinya Du, Sha Li, and Heng Ji. 2022. Dynamic global memory for document-level argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5264–5275, Dublin, Ireland. Association for Computational Linguistics.

Xinya Du, Alexander Rush, and Claire Cardie. 2021. GRIT: Generative role-filler transformers for document-level event entity extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 634–644, 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.

Guanhua Huang, Runxin Xu, Ying Zeng, Jiaze Chen, Zhouwang Yang, and Weinan E. 2023. An iteratively parallel generation method with the pre-filling strategy for document-level event extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10834–10852. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Dong-Ho Lee, Akshen Kadakia, Kangmin Tan, Mahak Agarwal, Xinyu Feng, Takashi Shibuya, Ryosuke Mitani, Toshiyuki Sekiya, Jay Pujara, and Xiang Ren. 2022. Good examples make a faster learner: Simple demonstration-based learning for low-resource NER. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2687–2700, Dublin, Ireland. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lei Li, Jing Chen, Bozhong Tian, and Ningyu Zhang. 2023. Revisiting k-NN for Fine-tuning Pre-trained Language Models. ArXiv:2304.09058 [cs].

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. Association for Computational Linguistics.

Wanlong Liu, Shaohuan Cheng, Dingyi Zeng, and Qu Hong. 2023. Enhancing document-level event argument extraction with contextual clues and role relevance. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12908–12922, Toronto, Canada. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for

extraction? PAIE: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6759–6774, Dublin, Ireland. Association for Computational Linguistics.

Yubing Ren, Yanan Cao, Fang Fang, Ping Guo, Zheng Lin, Wei Ma, and Yi Liu. 2022. CLIO: Role-interactive multi-event head attention network for document-level event extraction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2504–2514, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv e-prints*, pages arXiv–1904.

Xi'ao Su, Ran Wang, and Xinyu Dai. 2022. Contrastive learning-enhanced nearest neighbor mechanism for multi-label text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 672–679, Dublin, Ireland. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Quan Tran, Varun Manjunatha, Lidan Wang, Rajiv Jain, Doo Soon Kim, Walter Chang, and Thien Huu Nguyen. 2021. Inducing rich interaction structures between words for document-level event argument extraction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 703–715. Springer.

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, Peiyi Wang, Tianyu Liu, Shuang Zeng, Baobao Chang, and Zhifang Sui. 2022. 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, Seattle, United States. 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.

Yuqing Yang, Qipeng Guo, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2023. An AMR-based link prediction approach for document-level event argument extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12876–12889, Toronto, Canada. Association for Computational Linguistics.

Qi Zeng, Qiusi Zhan, and Heng Ji. 2022. EA^{2}E: Improving consistency with event awareness for document-level argument extraction. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2649–2655, Seattle, United States. Association for Computational Linguistics.

Hongxin Zhang, Yanzhe Zhang, Ruiyi Zhang, and Diyi Yang. 2022. Robustness of Demonstration-based Learning Under Limited Data Scenario. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1769–1782, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020. A two-step approach for implicit event argument detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7479–7485, Online. Association for Computational Linguistics.

Gang Zhao, Xiaocheng Gong, Xinjie Yang, Guanting Dong, Shudong Lu, and Si Li. 2023. DemoSG: Demonstration-enhanced schema-guided generation for low-resource event extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1805–1816, Singapore. Association for Computational Linguistics.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.

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.

A Model size and computational budget

Our experiments were powered by an NVIDIA GeForce RTX 4090 graphics card, boasting 24GB of GDDR6X VRAM, which offered ample memory for our deep learning tasks.

Our model training was conducted in two stage. Table 8 shows the parameters for MILE in the two-stage training. For our training, we opted for the $\text{BART}_{\text{large}}$ model because of its extensive parameterization and demonstrated effectiveness in advanced natural language processing tasks. In the first phase, the model underwent full training with a parameter size of 406M. In the second phase, the encoder parameters were fixed, and the model was trained with a parameter size of 203M.

Model	Training Stage 1	Training Stage 2
Mile	406M	203M

Table 8: Parameters of the two-stage training parameters

B Hyperparameters Set

To ensure the replicability of our findings, we meticulously recorded all pertinent training parameters, including learning rates and batch sizes. We used grid search to find the optimal parameters.

Hyperparameters	RAMS-1	RAMS-2	Wiki-1	Wikis-2
Learning rate	4e-5	5e-5	4e-5	6e-5
Batch size	16	56	16	32
Epochs	8	6	20	16
Max Sequence Length	512	512	512	512
Max Output Length	72	72	71	71
Weight Decay	1e-5	1e-5	1e-5	1e-5
Gradient Clip	1.0	1.0	1.0	1.0
Accumulate Grad Batches	6	12	2	4
LR Scheduler	linear	linear	linear	linear
Freeze encoder	True	False	True	False

Table 9: Hyperparameters for two-stage training.

The key hyperparameters were explored within the following search space:

1)The learning rate was searched within the range of [4e-5, 6e-5], incrementing by 1e-5. 2)The batch size was searched within the range of [8, 16] in training stage 1, incrementing by 2, and within the range of [32, 56] in training stage 2, incrementing by 16. 3) The number of epochs was searched within the range of [6, 12] in RAMS, incrementing by 1, and within the range of [12, 24] in Wikievents, incrementing by 2. 2)The accumulate grad batches was searched within the range of [1, 12], incrementing by 1. 2)The gradient clip batches was searched within the range of [1.0, 5.0], incrementing by 0.5. Table 9 shows the final selected values.