

# A Compressive Memory-based Retrieval Approach for Event Argument Extraction

Wanlong Liu<sup>1</sup>, Enqi Zhang<sup>1</sup>, Shaohuan Cheng<sup>1</sup>, Li Zhou<sup>2</sup>, Dingyi Zeng<sup>1</sup>,  
Chen Zhang<sup>3</sup>, Malu Zhang<sup>1</sup>, Wenyu Chen<sup>1\*</sup>

<sup>1</sup>University of Electronic Science and Technology of China

<sup>2</sup>The Chinese University of Hong Kong, Shenzhen

<sup>3</sup>National University of Singapore

liuwanlong@std.uestc.edu.cn, maluzhang@uestc.edu.cn, cw@uestc.edu.cn

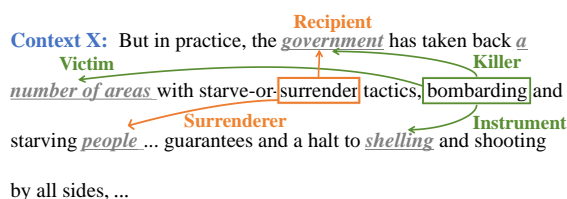
## Abstract

Recent works have demonstrated the effectiveness of retrieval augmentation in the Event Argument Extraction (EAE) task. However, existing retrieval-based EAE methods have two main limitations: (1) input length constraints and (2) the gap between the retriever and the inference model. These issues limit the diversity and quality of the retrieved information. In this paper, we propose a Compressive Memory-based Retrieval (CMR) mechanism for EAE, which addresses the two limitations mentioned above. Our compressive memory, designed as a dynamic matrix that effectively caches retrieved information and supports continuous updates, overcomes the limitations of the input length. Additionally, after pre-loading all candidate demonstrations into the compressive memory, the model further retrieves and filters relevant information from memory based on the input query, bridging the gap between the retriever and the inference model. Extensive experiments show that our method achieves new state-of-the-art performance on three public datasets (RAMS, WikiEvents, ACE05), significantly outperforming existing retrieval-based EAE methods.

## 1 Introduction

Event argument extraction (EAE) is a crucial and challenging subtask of information extraction (Ren et al., 2022b; Yang et al., 2021; Ding et al., 2024), aimed at identifying event-related arguments and determining their roles within texts. For instance, as shown in Figure 1, when the target event is `Life.die.death` with the trigger `bombarding`, EAE models are tasked with extracting arguments like “government” and “shelling”, which correspond to the roles of *attacker*, and *instrument*.

With the successful application of retrieval-augmented generation (RAG) (Lewis et al., 2020) technology to various NLP tasks (Levonian et al.,



Events
Event Type: <code>Life.die.death-caused-by-violent-events</code>
Template: <u>Killer</u> killed <u>victim</u> using <u>instrument</u> at place.
Event Type: <code>Life.die.conflict.yield.surrender</code>
Template: <u>Surrenderer</u> surrendered to <u>recipient</u> at place.

Figure 1: An example of an EAE task from the RAMS dataset (Ebner et al., 2020). Each underlined section in the template (prompt), known as a role slot, corresponds to a specific argument role.

2023; Yin et al., 2024b,a; Ni et al., 2024; Wang et al., 2024; Liang et al., 2024), some works (Du and Ji, 2022; Du et al., 2022; Ren et al., 2023; Huang, 2023) have incorporated retrieval-augmented techniques into event extraction. They use similarity-based retrieval to retrieve the most relevant instances (demonstrations) from the training set for the input query, providing prior external knowledge and augmenting the EAE process. However, these retrieval-based EAE methods still face some issues that hinder further improvement.

**First, retrieval augmentation is limited by the model input length.** Current mainstream generation-based EAE approaches typically utilize BART (Lewis et al., 2019) or T5 (Raffel et al., 2020) as the PLM. Consequently, due to the input length limitation of these inference models, only a very limited amount of retrieved information can be used for augmentation. For instance, in previous retrieval-based EAE methods (Ren et al., 2023; Huang, 2023), the number of retrieved demonstrations is limited to just one, which significantly limits the diversity of retrieved content.

\*Corresponding author

**Second, the retrieval quality is limited by the gap between the retriever and the inference model.** Current mainstream retrieval-based EAE methods (Ren et al., 2023; Huang, 2023; Du et al., 2022) use dense retrievers such as S-BERT (Reimers, 2019) and retrieve based on the similarity of the context. These retrievers, often untrained, exhibit an embedding gap with inference models as highlighted in recent studies (Ren et al., 2022a; Thakur et al., 2021; Xu et al., 2023), leading to sub-optimal retrieval quality. Additionally, in EAE tasks, only a few contextual words serve as event arguments, while other extraneous content can mislead the retriever, resulting in the retrieval of irrelevant demonstrations.

Recently, numerous studies (Munkhdalai et al., 2024; Katharopoulos et al., 2020; Tiezzi et al., 2024; Gu and Dao, 2023) have adopted RNN-inspired approaches to tackle the quadratic complexity issue of processing long texts in transformers. Inspired by these works, we propose a **Compressive Memory-based Retrieval (CMR)** method for EAE, which effectively addresses the two issues mentioned above. Specifically, we design a compressive memory mechanism that caches the information of retrieved demonstrations. This compressive memory, structured as a dynamic matrix, supports continuous updates and is theoretically capable of caching information indefinitely. Before inference, the model pre-loads all candidate demonstrations into the memory. Then it dynamically retrieves necessary information from the memory based on the input query, enabling adaptive filtering of the candidate demonstrations retrieved by the retriever.

Our proposed CMR model has the following two advantages over existing EAE methods: (1) CMR breaks the limitation of the model’s context window size, enabling the retrieval of more instances as demonstrations and ensuring the diversity of RAG. (2) CMR integrates a memory retrieval mechanism that further filters the retrieved information, allowing adaptive retrieval of relevant data for the EAE task. Its parameters are jointly trained with those of the PLM, bridging the gap between the retriever and inference model. Additionally, we introduce a training strategy that enhances the efficiency of the training process and improves the robustness of the model. Our contributions are summarized as follows:

- We propose a Compressive Memory-based

Retrieval (CMR) mechanism for EAE, employing a dynamic memory matrix to store retrieved demonstrations. This approach enables existing EAE models to handle larger volumes of retrieved content, significantly enhancing retrieval diversity.

- Our CMR mechanism can further filter retrieved information from candidate demonstrations, reducing interference from irrelevant information and bridging the gap between the retriever and inference model.
- Extensive experiments demonstrate that the proposed CMR mechanism outperforms previous retrieved-based EAE methods. Further experimental analysis demonstrates the effectiveness and robustness of our method.

## 2 Methodology

In this section, we first provide a formal definition of the EAE task. Consider an instance  $(X, \{e_i\}_{i=1}^K, \{t_i\}_{i=1}^K, \{R^{(e_i)}\}_{i=1}^K)$ , where  $X = \{w_0, w_1, \dots, w_{N-1}\}$  represents the document text consisting of  $N$  words, and  $K$  is the number of target events. Here,  $e_i$  denotes the type of the  $i$ -th event,  $t_i \subseteq X$  represents the trigger word of the  $i$ -th event, and  $R^{(e_i)}$  indicates the set of roles associated with the event  $e_i$ . The objective is to extract a set of spans  $\mathcal{S}_i$  for each event  $e_i$ , which satisfies  $\forall a^{(r)} \in \mathcal{S}_i, (a^{(r)} \subseteq X) \wedge (r \in R^{(e_i)})$ . Following this, we introduce the traditional RAG architecture for EAE and then describe our proposed Compressive Memory-based Retrieval (CMR) architecture.

### 2.1 Traditional RAG Architecture for EAE

Traditional retrieval-based EAE methods typically retrieve the demonstrations from a knowledge base, such as the training set. Specifically, when predicting the  $i$ -th event  $e_i$  in a document, the knowledge base is  $K = \{s_1, s_2, \dots, s_n\}$ , where  $s_i$  denotes the candidates to be retrieved<sup>1</sup>. Then, using S-BERT embeddings (Reimers, 2019), the cosine similarity between  $e_i$ ’s context  $c_i$  and each candidate in  $K$  is calculated, and the candidate with the highest score is selected as additional input to

<sup>1</sup>The candidate  $s_i$  can be the context (Ren et al., 2023) or event predictions (Du et al., 2022). In our implementation, we use both the context and event predictions of each instance as candidates (detailed in Section 2.2.)

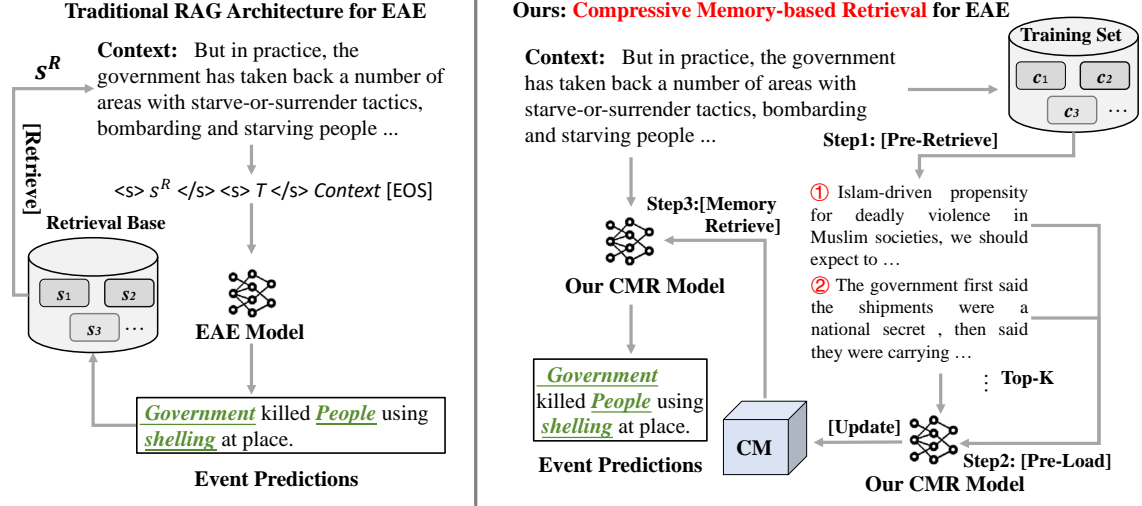


Figure 2: Overview of Compressive Memory-based Retrieval architecture. “CM” denotes the Compressive Memory. First, the model pre-loads all retrieved candidate demonstrations to build the memory. Then, it dynamically retrieves information from the memory based on the input query, and subsequently generates the final prediction.

enhance the prediction of  $e_i$ :

$$\text{score}(s_j, c_i) = \frac{\exp f(c_i, s_j)}{\sum_{s_j \in M} \exp f(c_i, s_j)},$$

$$f(c_i, s_j) = \text{S-BERT}(c_i)^T \text{S-BERT}(s_j), \quad (1)$$

$$s_i^R = \arg \max_{s_j} \text{score}(s_j, c_i),$$

where  $s_i^R$  denotes the retrieved candidate that  $e_i$  depends on. Then  $s_i^R$  is concatenated as a prefix to the input to enhance the model’s performance:

Input =  $\langle s \rangle s_i^R \langle /s \rangle \langle s \rangle P \langle /s \rangle x_1, x_2, \dots, x_N [\text{EOS}]$ .

where  $x_1, x_2, \dots, x_N$ , are the context words,  $\langle s \rangle$  and  $\langle /s \rangle$  denote special delimiter tokens, and  $P$  indicates the task prompt<sup>2</sup>.  $P$  and the context words form the event context  $c_i$ .

## 2.2 Compressive Memory-based Retrieval

Traditional retrieval-based EAE architectures primarily face two major issues: (1) Due to the input length limitation of PLMs, the retrieved content is restricted to the most similar candidate, and severely lacking in diversity. (2) The retriever uses fixed parameters and is not trained alongside the model to adapt to downstream tasks.

Inspired by Linear Attention mechanism (Katharopoulos et al., 2020), we introduce our Compressive Memory-based Retrieval (CMR) mechanism for EAE in this section. Our CMR mechanism addresses the above two issues: (1)

<sup>2</sup>Typically, it is an unfilled template (Ma et al., 2022; Huang, 2023) of the target event.

The CMR mechanism overcomes the limitation of model input length, theoretically enabling the retrieval of an unlimited number of demonstrations. (2) It incorporates a memory retrieval mechanism that can further filter the information, enabling the model to adaptively retrieve useful information for the EAE task. Its parameters are jointly trained with those of the PLM, bridging the gap between the retriever and inference model. In Appendix C, we prove that our CMR mechanism enables the information retrieval of demonstrations stored in memory.

**Compressive Memory.** We design a compressive memory  $M$  for each transformer layer to store candidate demonstrations encountered by the model. Unlike traditional vector retrieval databases, this memory is a fixed-size matrix. Each time the model finishes processing a candidate instance, the memory is updated based on the Key-Value (KV) cache of that instance. Note that the compressive memory is not part of the model parameters and can be inserted or removed as needed. When previous memories are no longer required,  $M$  can be reset to zero, effectively erasing all stored information.

**Memory Storage and Update.** For simplicity, we only illustrate the memory mechanism for a single layer. Given the context of the instance  $q$  and the retrieved demonstrations  $D = \{d_1, d_2, \dots, d_k\}$ , our CMR mechanism first stores these demonstrations into the compressive memory. To prevent memory overflow, inspired by (Katharopoulos et al., 2020), we introduce a normalization term

$\mathbf{n} \in \mathbb{R}^{d_k}$ , using a sum of all keys for normalization. For each demonstration  $d_i$ , represented by the embedding  $\mathbf{X}^{d_i} \in \mathbb{R}^{N \times d_{\text{model}}}$ , the memory and normalization term are updated as follows:

$$\mathbf{K}^{d_i} = \mathbf{X}^{d_i} \mathbf{W}_k, \mathbf{V}^{d_i} = \mathbf{X}^{d_i} \mathbf{W}_v, \quad (2)$$

$$\mathbf{M}_i \leftarrow \mathbf{M}_{i-1} + \sigma(\mathbf{K}^{d_i})^T \mathbf{V}^{d_i}, \quad (3)$$

$$\mathbf{n}_i \leftarrow \mathbf{n}_{i-1} + \sum_{j=1}^N \sigma(\mathbf{K}_j^{d_i}), \quad (4)$$

where  $\mathbf{W}_k \in \mathbb{R}^{d_{\text{model}} \times d_k}$  and  $\mathbf{W}_v \in \mathbb{R}^{d_{\text{model}} \times d_v}$  are trainable parameters from the transformer. Activation function  $\sigma$  is the element-wise ELU + 1 (Clev-ert et al., 2015) function.

**Memory Retrieval.** The process of memory retrieval is integrated into the transformer’s multi-head attention mechanism. For the instance  $q$ , represented by the embedding  $\mathbf{X} \in \mathbb{R}^{N \times d_{\text{model}}}$ , we initially calculate the vanilla dot-product attention (for a single head)  $\mathbf{A}_{\text{dot}} \in \mathbb{R}^{N \times d_v}$  as follows:

$$\mathbf{A}_{\text{dot}} = \text{softmax} \left( \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_{\text{model}}}} \right) \mathbf{V}, \quad (5)$$

$$\mathbf{K} = \mathbf{X}\mathbf{W}_k, \mathbf{V} = \mathbf{X}\mathbf{W}_v, \mathbf{Q} = \mathbf{X}\mathbf{W}_q. \quad (6)$$

Subsequently, we utilize the input query  $\mathbf{Q} \in \mathbb{R}^{N \times d_k}$  to retrieve from memory, obtaining the retrieval-augmented representation  $\mathbf{A}_{\text{ret}} \in \mathbb{R}^{N \times d_v}$ :

$$\mathbf{A}_{\text{ret}} = \frac{\sigma(\mathbf{Q})\mathbf{M}_k}{\sigma(\mathbf{Q})\mathbf{n}_k}. \quad (7)$$

Here,  $\mathbf{M}_k \in \mathbb{R}^{d_k \times d_v}$  is the compressive memory that stores information of all demonstrations, and  $\mathbf{n}_k \in \mathbb{R}^{d_k}$  is the normalization term, which is crucial for training stability.

Then, we combine the vanilla dot-product attention  $\mathbf{A}_{\text{dot}}$  and the retrieved  $\mathbf{A}_{\text{ret}}$  using a gating mechanism:

$$\mathbf{A} = S(\gamma) \odot \mathbf{A}_{\text{ret}} + (1 - S(\gamma)) \odot \mathbf{A}_{\text{dot}}, \quad (8)$$

where  $\gamma$  is a trainable gating scalar, and  $S(\cdot)$  denotes the Sigmoid function. Through the trainable gating scalar  $\gamma$ , the model achieves a learnable balance between input and retrieved information. Note that since the stored KV entries implicitly include the model’s predictions, our memory update process retains both the context of candidate demonstrations and the model’s event predictions.

## 2.3 Implementation

The proposed CMR mechanism can be well applied to both encoder-decoder and decoder-only architectures. (1) For models with an encoder-decoder architecture, the memory retrieval operations in Section 2.2 are implemented in the cross-attention module of each decoder layer, using the decoder’s input as  $\mathbf{Q}$  illustrated in Equation 7. (2) For decoder-only models, we replace the vanilla self-attention mechanism in each layer with our CMR mechanism.

### 2.3.1 Training

During the training process, we need to teach the model how to retrieve relevant information from memory to enhance generation for the EAE task. However, pre-retrieving the top-k-related candidate demonstrations for each training instance entails certain limitations: (1) The fixed number of retrieved demonstrations during training may restrict the model to a specific demonstration count, limiting the robustness of RAG. (2) Such a training approach is very time-consuming.

Therefore, we propose an efficient and robust training method. Specifically, we set a maximum retrieval number  $Max\_retrieval$  and initialize the memory  $\mathbf{M}_0$  to zero. Within  $Max\_retrieval$ , the model updates its memory as it infers each training instance<sup>3</sup>. When the number of instances stored in memory exceeds  $Max\_retrieval$ , the memory is reset to zero and the cycle repeats. The  $Max\_retrieval$  is set to match the model’s gradient accumulation steps. To ensure the relevance of the retrieved information, we rerank the shuffled training data in each epoch, organizing batches so that each training instance is primarily surrounded by instances of the same event type<sup>4</sup>, while also including a strategic mix of instances from different types to enhance model generalization and prevent overfitting. The detailed training algorithm is outlined in Algorithm 1 in Appendix A.

Our proposed training method has the following two advantages: (1) It significantly reduces training time. (2) Within each cycle of  $Max\_retrieval$ , the count of instances stored in memory continuously increases. This naturally provides training instances with varying retrieval numbers, which

<sup>3</sup>These stored instances will act as demonstrations for subsequent training instances.

<sup>4</sup>In EAE task, instances of the same event type often have high relevance to each other (Ebner et al., 2020; Huang, 2023).

enables the model to adapt to varying retrieval volumes, enhancing its robustness.

### 2.3.2 Inference

During inference, the model first pre-loads all candidate demonstrations to build memory. Specifically, each retrieved demonstration (context) is fed into the model, and the memory is updated according to Equations 3 and 4. Notably, during the pre-loading of each demonstration, the memory is only updated but does not participate in the attention calculation. To improve efficiency, we pre-load candidate demonstrations in batches, significantly reducing inference time.

Subsequently, the model dynamically retrieves necessary information from the memory based on the input query (context of the current inference instance), facilitating adaptive filtering of information from candidate demonstrations. As for the input order of candidate demonstrations, we illustrate in the experimental section that our model is not sensitive to the input order. The inference algorithm is detailed in Algorithm 2 in Appendix A.

## 3 Experiments

This section applies the proposed CMR mechanism to the current mainstream EAE baselines across three commonly used EAE benchmarks. Subsequently, we extend the CMR mechanism to decoder-only large language models to further explore its effectiveness. Additionally, we conduct detailed analytical experiments to analyze our method across various settings.

### 3.1 Experimental Setup

#### 3.1.1 Datasets

We conduct experiments on three widely used EAE datasets: RAMS (Ebner et al., 2020), WikiEvents (Li et al., 2021), and ACE2005 (Dodgington et al., 2004). Detailed descriptions of these datasets are provided in Appendix B.1.

#### 3.1.2 Baselines

We categorize the baselines for comparison into two groups: **W.o. Retrieval** and **With Retrieval**.

**W.o. Retrieval:** We select recent state-of-the-art EAE methods, including DEEIA (Liu et al., 2024), TabEAE (He et al., 2023), SPEAE (Nguyen, 2023), SCPRG (Liu et al., 2023), PAIE (Ma et al., 2022), and BART-Gen (Li et al., 2021).

**With Retrieval:** We choose some classic retrieval-based EAE methods, including R-GQA (Du and Ji,

2022) and AHR (Ren et al., 2023). Since previous retrieval-based EAE methods did not use uniform datasets and metrics for evaluation, to ensure a more comprehensive and fair comparison, we establish two retrieval-based EAE baselines PAIE-R and BART-Gen-R based on two commonly used methods, PAIE and BART-Gen. Specifically, we follow (Du and Ji, 2022), using the S-BERT retriever to identify and incorporate the most relevant (Top-1) event prediction as a prefix into the input.

#### 3.1.3 Evaluation Metrics

Following earlier studies (Ma et al., 2022; He et al., 2023), we evaluate the performance using two metrics: (1) Argument Identification F1 (Arg-I), which deems a predicted event argument correct if its boundaries align with any corresponding reference arguments. (2) Argument Classification F1 (Arg-C), requiring both boundary and role type accuracy for a predicted event argument to be considered correct. Our experiments are conducted five times with different seeds, and we report the average results.

### 3.2 Main Results

**Comparison with W.o. Retrieval methods.** As shown in Table 1, our PAIE-CMR and BART-Gen-CMR models outperform previous non-retrieval SOTA methods, such as SCPRG and DEEIA, showcasing a strong competitive advantage.

**Comparison with Retrieval-based methods.** As shown in Table 1, two classic EAE baselines, PAIE and BART-Gen, achieve improved performance across all three datasets after incorporating retrieval, which highlights the positive impact of RAG on the EAE task. However, the performance improvement of PAIE-R and BART-Gen-R over the baseline is minimal, demonstrating the limitations of previous retrieval-based EAE methods. These methods are restricted to retrieving only the top-1 demonstration, which severely lacks diversity and results in sub-optimal performance. In contrast, our CMR mechanism ensures the diversity of retrieved demonstrations and further filters the information, achieving superior performance.

### 3.3 CMR for Decoder-Only LLMs

In this section, we explore the effectiveness of our CMR mechanism on decoder-only LLMs. We fine-tune LLaMA3-8b-instruct (Touvron et al., 2023) on the RAMS dataset and evaluate the performance of our method.

Scheme	Method	PLM	RAMS		WikiEvents		ACE2005	
			Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
<i>W.o. Retrieval</i>	DEEIA (2024)	RoBERTa-l	<u>58.0</u>	<u>53.4</u>	<u>71.8</u>	<u>67.0</u>	76.3	74.1
	TabEAE (2023)	RoBERTa-l	57.3	52.7	71.4	66.5	<b>77.2</b>	<b>75.0</b>
	SPEAE (2023)	BART-l	58.0	53.3	71.9	66.1	-	-
	SCPRG (2023)	RoBERTa-l	56.7	52.3	71.3	66.4	-	-
	PAIE (2022)	BART-l	56.8	52.2	70.5	65.3	72.1	70.8
	BART-Gen (2021)	BART-l	51.2	48.6	66.8	62.4	69.9	66.7
<i>With Retrieval</i>	R-GQA (2022)	BART-l	-	-	-	-	75.5	72.8
	AHR (2023)	T5-l	54.6	48.4	69.6	63.4	-	-
	PAIE-R*	BART-l	57.4	53.0	71.2	66.0	73.0	71.9
	BART-Gen-R*	BART-l	51.4	49.1	67.9	63.2	70.2	66.9
	PAIE-CMR (Ours)	BART-l	<b>59.1</b> ( $\uparrow 1.7$ )	<b>54.3</b> ( $\uparrow 1.3$ )	<b>72.8</b> ( $\uparrow 1.6$ )	<b>67.9</b> ( $\uparrow 1.9$ )	<u>76.8</u> ( $\uparrow 3.8$ )	<u>74.8</u> ( $\uparrow 2.9$ )
	BART-Gen-CMR (Ours)	BART-l	53.2 ( $\uparrow 1.8$ )	51.4 ( $\uparrow 2.3$ )	69.1 ( $\uparrow 1.2$ )	65.3 ( $\uparrow 2.1$ )	<u>72.4</u> ( $\uparrow 2.2$ )	<u>69.3</u> ( $\uparrow 2.4$ )

Table 1: Comparison of performance on RAMS, WikiEvents, and ACE2005 test set. \* means that we add traditional retrieval into the original method. The shaded area represents our methods, which retrieve top-10 demonstrations. **Bold** and underline indicate the best and second-best experimental results.

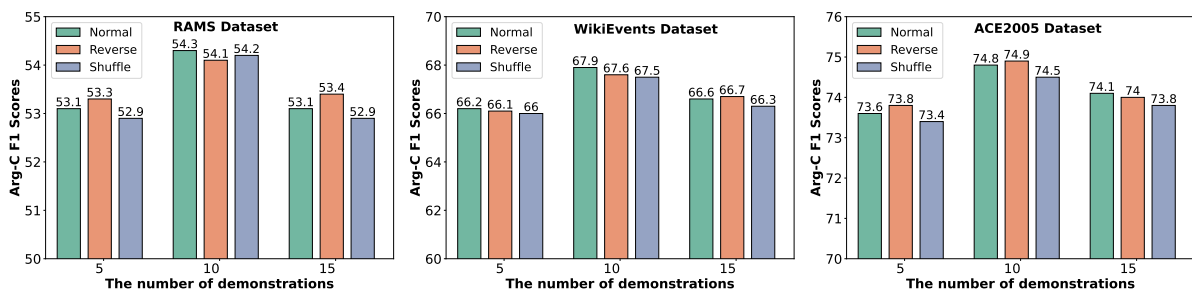


Figure 3: Demonstrations order experiment for PAIE-CMR. **Normal** uses the top-k demonstrations in their original retrieved order, **Reverse** uses them in the opposite order, and **Shuffle** means randomly shuffling the demonstrations.

Method	#N	RAMS		WikiEvents		ACE2005	
		Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
PAIE-CMR	0	56.1	51.8	70.6	64.8	71.9	70.4
	1	57.8	53.1	71.4	66.2	74.3	72.9
	5	58.5	53.6	72.0	66.9	75.2	73.6
	10	<b>59.1</b>	<b>54.3</b>	<b>72.8</b>	<b>67.9</b>	<b>76.8</b>	<b>74.8</b>
	15	58.8	54.0	72.4	67.5	76.2	74.1
B-G-CMR	0	51.0	47.5	66.0	61.7	69.2	66.4
	1	52.0	49.9	68.4	64.1	70.8	67.7
	5	52.7	50.8	<b>69.1</b>	<b>65.3</b>	72.0	69.1
	10	<b>53.2</b>	<b>51.4</b>	68.9	64.7	<b>72.4</b>	<b>69.3</b>
	15	52.9	51.0	68.5	64.1	71.7	68.6

Table 2: The performance of retrieving varying numbers of demonstrations in CMR mechanism. #N represents the number of retrieved top-k demonstrations, with #N equals to 0 indicating no retrieval.

**Evaluation Metrics.** We establish two evaluation metrics to evaluate the performance of the LLM-based EAE models: (1) **Strict-F1**, which considers a predicted event argument correct if the model’s prediction exactly matches the golden label. (2) **Relaxed-F1**, which considers a prediction correct if the golden label is contained within the model’s prediction.

**Experimental Details.** We select LLaMA3-8b-instruct for full-parameter fine-tuning on RAMS

training set and evaluate it on the RAMS, WikiEvents, and ACE2005 test sets. First, we train LLaMA3-SFT-CMR using the CMR mechanism, following the training strategy outlined in Section 2.3.1. For comparison, we also train a LLaMA3-SFT model using standard supervised fine-tuning. The inference process follows Algorithm 2 in appendix A. Additional training details, including prompts and experimental settings, are provided in Appendix B.3.

**Analysis.** As shown in Table 3: (1) For LLaMA3-SFT, the impact of RAG after supervised fine-tuning is minimal, even with a decline in performance on WikiEvents dataset. (2) In contrast, our LLaMA3-SFT-CMR model performs better on all three datasets, underscoring the effectiveness of our CMR mechanism in decoder-only LLM architectures and demonstrating the generalizability of our approach. (3) However, the overall improvement of LLaMA3-SFT-CMR over LLaMA3-SFT remains limited. We assume that this is due to the large parameter size of the model, combined with the relatively small size and limited task diversity of the fine-tuning data, which may hinder the model’s ability to fully learn the CMR capability.

Method	#N	RAMS		WikiEvents		ACE2005	
		Strict F1	Relaxed F1	Strict F1	Relaxed F1	Strict F1	Relaxed F1
LLaMA3-SFT-CMR	5	32.48	35.25	23.11	31.63	30.72	42.97
LLaMA3-SFT-CMR	10	32.78	37.24	<b>24.22</b>	<b>32.96</b>	<b>31.21</b>	<b>43.54</b>
LLaMA3-SFT	0	31.05	34.93	22.41	30.99	27.81	40.28
LLaMA3-SFT	5	31.65	35.63	22.77	31.00	28.88	41.69
LLaMA3-SFT	10	32.96	36.10	21.23	30.64	29.49	41.35
LLaMA3-SFT-R	0	29.74	32.68	19.43	29.67	28.95	39.40
LLaMA3-SFT-R	5	32.72	35.34	22.02	31.08	30.46	41.66
LLaMA3-SFT-R	10	<b>33.08</b>	<b>37.42</b>	22.80	31.63	30.28	41.87

Table 3: Performance comparison of models fine-tuned on LLaMA3-8b-instruct. LLaMA3-8b-SFT, LLaMA3-8b-SFT and LLaMA3-8b-SFT-R are all trained on the RAMS training set and then evaluated on the RAMS, WikiEvents, and ACE2005 test sets. #N indicates the number of retrieved demonstrations from corresponding training set. **Bold** highlights the best experimental results.

## 4 Analysis

In this section, we further analyze our CMR mechanism by addressing the following questions: Q1: How does the CMR mechanism compare to directly using a long-context model? Q2: How does the number of demonstrations during inference affect performance? Q3: What impact does the order of demonstrations have on performance? Q4: Can this method filter out irrelevant information and enhance the robustness of the RAG?

### 4.1 Q1: Compare with Long-Context Models

To evaluate the effectiveness of the CMR mechanism compared to directly using a long-context model, we select LLaMA3-8b-instruct as the base model and train **LLaMA3-SFT-R** model through retrieval-based training. Aligning with the training process of our CMR mechanism, we retrieve top 8 demonstrations for each training instance and insert these demonstrations into the prompt in Figure 4. The remaining fine-tuning details are consistent with those of LLaMA3-SFT.

As shown in Table 3, LLaMA3-SFT-R significantly improves performance over the non-retrieval scenario with retrieval. Additionally, although LLaMA3-SFT-R performs well on the RAMS dataset, it generalizes poorly to WikiEvents and ACE2005 when compared to our LLaMA3-SFT-CMR model. This suggests that simply using a long-context model to directly train RAG capabilities for EAE results in poor generalization. In contrast, our model learns to adaptively retrieve and filter information from memory during training, which enhances the generalization capability.

### 4.2 Q2: Analysis on Demonstration Numbers

Table 3 shows the performance of PAIE-CMR and BART-Gen-CMR across different numbers

Method	#N	#Mode	RAMS		WikiEvent	
			Arg-I	Arg-C	Arg-I	Arg-C
PAIE	0	No Ret.	56.8	52.2	70.5	65.3
PAIE-R	1	Top-k	57.4	53.0	71.2	66.0
	1	Random	56.2	51.5	70.1	64.4
PAIE-CMR	1	Top-k	57.6	53.1	71.4	66.2
	1	Random	57.2	52.5	70.6	65.6
	5	Top-k	58.5	53.6	72.0	66.9
	5	Random	57.7	53.1	71.4	66.6

Table 4: Experiments on retrieval robustness. We compare PAIE-R with our PAIE-CMR, highlighting the robustness of our retrieval method. #Mode={No Retrieval, Top-k Retrieval, Random Retrieval} represents the different retrieval modes. Random retrieval involves randomly selecting demonstrations from the training set.

of demonstrations. (1) When #N is 1, our CMR approach outperforms PAIE-R and BART-Gen-R. This improvement can be attributed to two reasons: (a) Our method uses more comprehensive demonstrations, including both context and implicit event predictions. (b) Our CMR mechanism adaptively filters retrieved information, reducing interference from irrelevant data. (2) As #N increases, the performance shows an improving trend across all three datasets. It suggests that the growing amount and diversity of retrieved information contributes to enhanced performance. Furthermore, it demonstrates that our CMR mechanism effectively stores information from candidate demonstrations and retrieves useful information efficiently. (3) However, when #N exceeds 10, the performance declines. We attribute this to the number of retrieved demonstrations surpassing the training limit of *Max\_retrieval*, making it difficult for the model to effectively store and manage the excessive information.

### 4.3 Q3: Analysis on Demonstration Order

To explore our method’s sensitivity to the order of demonstrations, we design three types of input

orders—**Normal**, **Reverse**, and **Shuffle**—and conduct inference on trained checkpoints from three datasets, respectively. We first retrieve the top-k demonstrations and then conduct inference using the PAIE-CMR in the aforementioned three orders. As illustrated in Figure 4, when the number of demonstrations is held constant, the performance of the three orders exhibits negligible variation, indicating that our method is insensitive to the order of demonstrations. We assume this is due to (a) the inference process ensuring that each demonstration’s memory remains unaffected by others, making the CMR mechanism order-insensitive, and (b) the instance shuffling during training across epochs, which further enhances our model’s robustness.

#### 4.4 Q4: Retrieval Robustness Analysis

To explore the retrieval robustness of our method, we implement two retrieval strategies: (1) **Topk**, which retrieves the top-k most similar demonstrations. (2) **Random**, which selects demonstrations randomly from the training set. As shown in Table 4, the traditional retrieval-based EAE method, PAIE-R, is highly sensitive to the relevance of the retrieved content. Its performance declines significantly with random retrieval, even dropping below that of using no retrieval at all. In contrast, our CMR mechanism demonstrates stronger robustness under conditions of random retrieval. It can be attributed to our training strategy, where we maintain a selection of unrelated demonstrations in memory during each gradient update. Furthermore, our CMR mechanism adaptively filters out irrelevant information, effectively reducing interference from noisy data. These two strategies significantly enhance the robustness of our model’s RAG robustness against irrelevant demonstrations.

In Appendix B.4, we also conduct experiments to evaluate our model’s performance with RAG under new ontologies, demonstrating its robust generalizability across domain transfer scenarios.

## 5 Related Works

### 5.1 Event Argument Extraction

Event argument extraction (EAE) aims to extract specific details about the identified events, such as their locations or the individuals involved, which is a challenging task in Natural Language processing (Yu et al., 2021, 2024). Recent mainstream EAE methods can be primarily divided into the following two categories. (1) Span-based methods,

which identify candidate spans and predict their roles (Zhang et al., 2020; Yang et al., 2023; Liu et al., 2017; Zhang et al., 2020; Liu et al., 2023; Xu et al., 2022). (2) Generation-based methods, which have recently gained popularity, utilize slotted templates and a generative slot-filling strategy for argument extraction (Ma et al., 2022; He et al., 2023; Nguyen, 2023; Li et al., 2021; Huang, 2023; Zeng et al., 2022). While both methods offer distinct advantages, generation-based methods have demonstrated superior generalizability and competitive performance compared to their span-based counterparts (Hsu et al., 2023).

With the advancement of RAG technology (Lewis et al., 2020), some works (Du and Ji, 2022; Ren et al., 2023; Huang, 2023) have incorporated RAG techniques into event extraction, leading to some performance boosts. However, these methods are constrained by the model’s input length, resulting in a limited amount of content available for retrieval enhancement, which significantly restricts both the diversity and quality of RAG. These methods also suffer from a substantial information gap between the retriever and the inference model, which leads to sub-optimal performance.

### 5.2 RNN-Inspired Memory Methods for Transformers

Recently, numerous studies have adopted RNN-inspired approaches to tackle the quadratic complexity issue of processing long texts in transformers. For example, (Katharopoulos et al., 2020) introduces Linear Attention, which reduces complexity by efficiently retaining relevant information. Similarly, (Munkhdalai et al., 2024) proposes the Infinite Transformer, which utilizes the memory mechanism to allow the model to focus on previously stored information. Additionally, Mamba (Gu and Dao, 2023) incorporates memory-augmented attention, storing crucial past information for future reference. (Tiezzi et al., 2024) leverages state-space models to manage long-range dependencies. Inspired by these works, we propose a compressive memory mechanism that adaptively retrieves and dynamically updates stored information.

## 6 Conclusion

In this paper, we introduce a Compressive Memory-based Retrieval (CMR) mechanism to overcome input length constraints and the gap between the retriever and inference model in retrieval-



based EAE methods. Our approach uses a dynamic, updating matrix to continuously store demonstrations. By pre-loading candidate demonstrations and dynamically filtering based on the input query, our model significantly enhances the retrieval quality. Extensive experiments on three public datasets demonstrate that our method achieves new state-of-the-art performance, outperforming existing retrieval-based EAE methods.

## 7 Limitations

The improvement of our CMR mechanism when applied to LLM models (LLaMA3-8b-instruct) is limited. We hypothesize that this is due to the substantial number of model parameters, coupled with the relatively small scale and limited diversity of our training data. Many studies (Chen et al., 2024; Chung et al., 2024; Zhang et al., 2024) have emphasized that the diversity and quality of data are crucial for enhancing model performance. However, the event argument extraction data we currently use for training is inconsistent in quality and limited in quantity. Moving forward, we plan to explore the use of larger-scale or synthetic data and a more diverse set of tasks, aiming to extend our CMR mechanism to a broader range of NLP applications, including generative tasks such as question answering.

## Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under grant U22B2061. We would like to express our sincere gratitude to Qiao Liu for his significant contributions to this project. Qiao Liu received the PhD degree from University of Electronic Science and Technology of China(UESTC) in 2010. He was a visiting scholar at the Computer and Information Science Department of University of Pennsylvania from 2007 to 2009. He joined the School of Information and Software Engineering at UESTC as an associate professor in 2012. Now he serves as a professor at the School of Computer Science and Engineering of UESTC. His research interests include Natural Language Processing and Data Mining.

## References

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. 2024. How far are we to gpt-4v? closing the gap to commercial

multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Zhuojun Ding, Wei Wei, Xiaoye Qu, and Danyang Chen. 2024. Improving pseudo labels with global-local denoising framework for cross-lingual named entity recognition. *arXiv preprint arXiv:2406.01213*.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

Xinya Du and Heng Ji. 2022. Retrieval-augmented generative question answering for event argument extraction. *arXiv preprint arXiv:2211.07067*.

Xinya Du, Sha Li, and Heng Ji. 2022. Dynamic global memory for document-level argument extraction. *arXiv preprint arXiv:2209.08679*.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proc. of ACL*.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Yuxin He, Jingyue Hu, and Buzhou Tang. 2023. Revisiting event argument extraction: Can eae models learn better when being aware of event co-occurrences? *arXiv preprint arXiv:2306.00502*.

I Hsu, Zhiyu Xie, Kuan-Hao Huang, Prem Natarajan, Nanyun Peng, et al. 2023. Ampere: Amr-aware prefix for generation-based event argument extraction model. *arXiv preprint arXiv:2305.16734*.

Huang. 2023. From simple to complex: A progressive framework for document-level informative argument extraction. *arXiv preprint arXiv:2310.16358*.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR.

Zachary Levonian, Chenglu Li, Wangda Zhu, Anoushka Gade, Owen Henkel, Millie-Ellen Postle, and Wanli Xing. 2023. Retrieval-augmented generation to improve math question-answering: Trade-offs between groundedness and human preference. *arXiv preprint arXiv:2310.03184*.

- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.
- Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proc. of NAACL*.
- Qiuyu Liang, Weihua Wang, Feilong Bao, and Guanglai Gao. 2024. L<sup>2</sup>gc: Lorentzian linear graph convolutional networks for node classification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9988–9998.
- Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proc. of ACL*.
- 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.
- Wanlong Liu, Li Zhou, Dingyi Zeng, Yichen Xiao, Shaohuan Cheng, Chen Zhang, Grandee Lee, Malu Zhang, and Wenyu Chen. 2024. Beyond single-event extraction: Towards efficient document-level multi-event argument extraction. *arXiv preprint arXiv:2405.01884*.
- 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. *arXiv preprint arXiv:2202.12109*.
- Tsendsuren Munkhdalai, Manaal Faruqi, and Siddharth Gopal. 2024. Leave no context behind: Efficient infinite context transformers with infini-attention. *arXiv preprint arXiv:2404.07143*.
- Thien Nguyen, Chien. 2023. Contextualized soft prompts for extraction of event arguments. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4352–4361.
- Haowei Ni, Shuchen Meng, Xieming Geng, Panfeng Li, Zhuoying Li, Xupeng Chen, Xiaotong Wang, and Shiyao Zhang. 2024. Time series modeling for heart rate prediction: From arima to transformers. *arXiv preprint arXiv:2406.12199*.
- 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. *Journal of machine learning research*, 21(140):1–67.
- Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qifei Wu, Yuchen Ding, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2022a. A thorough examination on zero-shot dense retrieval. *arXiv preprint arXiv:2204.12755*.
- Yubing Ren, Yanan Cao, Fang Fang, Ping Guo, Zheng Lin, Wei Ma, and Yi Liu. 2022b. Clío: 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.
- 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.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.
- Matteo Tiezzi, Michele Casoni, Alessandro Betti, Marco Gori, and Stefano Melacci. 2024. State-space modeling in long sequence processing: A survey on recurrence in the transformer era. *arXiv preprint arXiv:2406.09062*.
- H Touvron, T Lavril, G Izacard, X Martinet, MA Lachaux, T Lacroix, B Rozière, N Goyal, E Hambro, F Azhar, et al. 2023. Open and efficient foundation language models. *Preprint at arXiv. https://doi.org/10.48550/arXiv.2302*.
- Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Transformer dissection: a unified understanding of transformer’s attention via the lens of kernel. *arXiv preprint arXiv:1908.11775*.
- Cunda Wang, Weihua Wang, Qiuyu Liang, Jie Yu, and Guanglai Gao. 2024. Gsea: Global structure-aware graph neural networks for entity alignment. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 187–199. Springer.
- 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. *arXiv e-prints*.

- Shicheng Xu, Liang Pang, Huawei Shen, and Xueqi Cheng. 2023. Berm: Training the balanced and extractable representation for matching to improve generalization ability of dense retrieval. *arXiv preprint arXiv:2305.11052*.
- 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.
- 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. *arXiv preprint arXiv:2305.19162*.
- Xin Yin, Chao Ni, Tien N Nguyen, Shaohua Wang, and Xiaohu Yang. 2024a. Rectifier: Code translation with corrector via llms. *arXiv preprint arXiv:2407.07472*.
- Xin Yin, Chao Ni, Shaohua Wang, Zhenhao Li, Limin Zeng, and Xiaohu Yang. 2024b. Thinkrepair: Self-directed automated program repair. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 1274–1286.
- Peiyang Yu, Victor Y Cui, and Jiaxin Guan. 2021. Text classification by using natural language processing. In *Journal of Physics: Conference Series*, volume 1802, page 042010. IOP Publishing.
- Peiyang Yu, Xiaochuan Xu, and Jiani Wang. 2024. Applications of large language models in multimodal learning. *Journal of Computer Technology and Applied Mathematics*, 1(4):108–116.
- Qi Zeng, Qiusi Zhan, and Heng Ji. 2022. Improving consistency with event awareness for document-level argument extraction. *arXiv preprint arXiv:2205.14847*.
- Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*.
- Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020. A two-step approach for implicit event argument detection. In *Proc. of ACL*.

## A Training and Inference Details

We propose an efficient and robust training method, and the detailed algorithm is shown in Algorithm 1. For clarity, we only describe the memory update process. Details on normalization and other operations can be found in Section 2.2 of the main text. The `ShuffleRerank` function first shuffles the training data to eliminate sequence-based patterns, promoting model generalization. After shuffling, the data is reranked by event type, ensuring each batch primarily contains instances of the same event type, with a strategic mix of 20% different types included to further enhance generalization and prevent overfitting. In the training process, data within each batch is processed in parallel.

---

### Algorithm 1 Efficient Training of CMR

**Require:** Training data  $T = \{s_1, s_2, \dots, s_n\}$ ,  
Maximum retrieval number  $Max\_retrieval$ ,  
Model  $\mathcal{M}$

**Ensure:** Trained model  $\mathcal{M}$

```
1:  $\mathbf{M}_0 \leftarrow \mathbf{0}, t \leftarrow 1$ 
2: for epoch  $e = 1$  to  $E$  do
3:    $\mathbf{D}_e \leftarrow \text{ShuffleRerank}(T)$  // Shuffle and
   rerank by event type
4:   for batch  $b \subset \mathbf{D}_e$  do
5:     for instance  $s_i \in b$  do
6:        $\mathbf{O}_t, \mathbf{M}_t \leftarrow \mathcal{M}(\mathbf{M}_{t-1}, s_i)$  // Forward
       propagate,  $\mathbf{O}_t$  denotes the event predic-
       tions of the model.
7:        $t \leftarrow t + 1$ 
8:     end for
9:      $\mathbf{M}_t \leftarrow \mathbf{M}_{t-|b|} + \frac{1}{|b|} \sum_{i=1}^{|b|} \mathbf{M}_{t-|b|+i}$  // Up-
     date memory
10:    if  $t > Max\_retrieval$  then
11:       $\mathbf{M}_0 \leftarrow \mathbf{0}, t \leftarrow 1$  // Reset memory and
      counter
12:      Update model parameters of  $\mathcal{M}$ 
13:    end if
14:  end for
15: end for
```

---

The detailed inference process is shown in Algorithm 2. `RetrieveTopK` uses S-BERT to retrieve the top-k relevant demonstration contexts based on similarity. During inference, data within each demonstration batch  $B_j$  is processed in parallel (as seen in lines 4-6 of Algorithm 2), significantly improving inference efficiency.

---

### Algorithm 2 Inference with CMR

**Require:** Knowledge base  $K$ , Input query  $q$ ,  
Model  $\mathcal{M}$ , Retrieval number  $k$

**Ensure:** Inference result for query  $q$

```
1:  $D \leftarrow \text{RetrieveTopK}(K, q, k)$  // Top- $k$  demon-
   strations
2:  $\mathbf{M}_0 \leftarrow \mathbf{0}, t \leftarrow 1$  // Initialize memory
3: for each batch  $B_j \subset D$  do
4:   for each  $d_i \in B_j$  do
5:      $\mathbf{M}_t \leftarrow \mathcal{M}(\mathbf{M}_0, d_i)$ 
6:      $t \leftarrow t + 1$ 
7:   end for
8:    $\mathbf{M}_t \leftarrow \mathbf{M}_{t-|B_j|} + \frac{1}{|B_j|} \sum_{i=1}^{|B_j|} \mathbf{M}_{t-|B_j|+i}$  //
   Update memory
9: end for
10:  $output \leftarrow \mathcal{M}(\mathbf{M}_k, q)$  // Final inference with
   memory and query
11: return  $output$ 
```

---

## B Experimental Analysis

### B.1 Dataset Statistics

We evaluate our proposed method on three event argument extraction (EAE) datasets.

**RAMS** (Ebner et al., 2020) is a document-level EAE dataset comprising 9,124 annotated events from English online news. We use a sliding window approach to aggregate events within the same context into single instances with multiple events, following the original train/dev/test split as in (He et al., 2023).

**WikiEvents** (Zhang et al., 2020) is a document-level EAE dataset with events from English Wikipedia and related news articles. Although it includes co-reference links for arguments, we only utilize the exact argument annotations in our experiments.

**ACE05** (Doddington et al., 2004) is a labeled corpus for information extraction, including newswire, broadcast news, and telephone conversations. We use the English event annotations for sentence-level EAE, following the preprocessing method described by (Ma et al., 2022).

The detailed dataset statistics for the three datasets are presented in Table 5.

### B.2 Implement Details for models in Encoder-Decoder Architecture

Our models, including PAIE-R, BART-Gen-R, PAIE-CMR and BART-Gen-CMR, based on encoder-decoder architectures, are run on a single

Dataset	RAMS	WikiEvents	ACE2005
# Event Types	139	50	33
# Events per Doc	1.25	1.78	1.19
# Args per Event	2.33	1.40	1.35
<b># Total Events</b>			
Training Set	7329	3241	4202
Validation Set	924	345	450
Test Set	871	365	403

Table 5: Overview of Dataset Statistics.

RTX 4090 GPU. All experimental results are averaged over five random seeds. The trainable gating scalar  $\gamma$  is initialized to 0 for all layers. The detailed hyperparameters for PAIE-CMR and BART-Gen-CMR are presented in Table 6 and Table 7.

Hyperparameters	RAMS	Wiki	ACE2005
Training Steps*	20000	20000	15000
Warmup Ratio	0.1	0.1	0.2
Learning Rate	2e-5	2e-5	2e-5
Gradient Accum Steps*	8	8	8
<i>Max_retrieval*</i>	8	8	8
Batch Size	4	4	16
Context Window Size	250	250	250
Max Span Length	10	10	10
Max Encoder Seq Length	500	500	500
Max Prompt Length	210	210	80
Demonstration Batch Size*	4	4	4

Table 6: Hyperparameter settings for PAIE-CMR. \* means that we tuned the hyperparameters in our experiments. The rest of hyperparameters are set the same as PAIE (Ma et al., 2022).

### B.3 Implement Details for Models in Decoder-Only Architecture

We choose LLaMA3-8b-instruct for full-parameter fine-tuning on the RAMS dataset. The experiments are conducted using four 80GB A100 GPUs, with training lasting approximately one hour for 3 epochs. The batch size is set to 2 per GPU, with 8 gradient accumulation steps, and the maximum input length is 4096 tokens. During the training process, we format the inputs as `<bos> X Y <eos>` and the labels as `<ignore> ...<ignore> Y <eos>`. In this setup, `<bos>` marks the beginning of the sequence, `X Y` represents the input context and label, and `<eos>` indicates the end of the sequence. The labels are structured to ignore the initial part of the

Hyperparameters	RAMS	Wiki	ACE2005
Training Epochs*	8	8	5
Warmup Ratio	0.0	0.0	0.0
Learning Rate	3e-5	3e-5	3e-5
Gradient Accum Steps*	8	8	8
<i>Max_retrieval*</i>	8	8	8
Batch Size	2	2	8
Weight Decay	0	0	0
Demonstration Batch Size*	4	4	4

Table 7: Hyperparameter settings for BART-Gen-CMR. \* means that we tuned the hyperparameters in our experiments. The rest of hyperparameters are set the same as PAIE (Huang, 2023).

sequence (denoted by `<ignore>` tokens), focusing only on `Y <eos>` for loss calculation during training. The prompts are specifically designed for the EAE task, as detailed in Figure 4 and Figure 5. We train the LLaMA3-SFT-CMR model using the CMR mechanism, following the training strategy in Section 2.3.1. The memory is updated only after the model processes an entire instance. For comparison, we also train a LLaMA3-SFT model using standard supervised fine-tuning.

### B.4 Domain Transfer Experiments

In this section, to simulate a real-world scenario, we explore the capabilities of our model with RAG applied to test sets of new ontologies (event types and argument types), following the studies by (Li et al., 2021; Du and Ji, 2022). Specifically, we conduct experiments on the RAMS, WikiEvents, and Ace05 datasets, training the model on the source dataset (`src`) and evaluating it on the target dataset (`tgt`). As shown in Table 8, compared to PAIE, our PAIE-CMR performs better in all domain transfer scenarios, demonstrating our model’s capability with RAG under new ontologies. This illustrates the robust generalizability of our approach.

## C Detailed Analysis of Compressive Memory-based Retrieval

In this section, we further analyze our CMR mechanism and show that it enables the retrieval of information from demonstrations stored in memory. First, we briefly introduce the concept of traditional attention and linear attention (Katharopoulos et al., 2020) to lay the groundwork for our approach, then we demonstrate that our approach can be considered a natural extension of linear attention and

**Task Description:** Given a document and an event, identify all arguments related to the event and classify the role of each argument. Limit your response to arguments only. Please provide your answer in the following JSON format: [{"type": "<role>", "argument": "<argument>"}, {"type": "<role>", "argument": "<argument>"}, ...].

**Retrieved Context:** <Provide the retrieved context here>

**Context:** <Provide the context that requires inference>

**Arguments:**

Figure 4: Our designed prompt for EAE task for normal decoder-only LLMs.

**Task Description:** Given a document and an event, identify all arguments related to the event and classify the role of each argument. Limit your response to arguments only. Please provide your answer in the following JSON format: [{"type": "<role>", "argument": "<argument>"}, {"type": "<role>", "argument": "<argument>"}, ...].

**Context:** <Provide the context that requires inference>

**Arguments:**

Figure 5: Our designed prompt for EAE task for our CMR-based LLMs.

therefore can be seen as a retrieval process of information from demonstrations stored in memory.

For an embedded input sequence  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ , traditional attention mechanism generates a sequence-to-sequence mapping by calculating the interactions between inputs from each location and inputs from other locations and integrating them into its own representation, obtaining the output sequences  $(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N)$ . Taking the  $i$ -th token as an example, and disregarding the scaling factor, the resulting output  $\mathbf{y}_i$  of the aggregated global information is as follows:

$$\mathbf{y}_i = \frac{\sum_{j=1}^N \exp(\mathbf{q}_i \mathbf{k}_j^T) \mathbf{v}_j}{\sum_{j=1}^N \exp(\mathbf{q}_i \mathbf{k}_j^T)}.$$

Here,  $\mathbf{q}_i, \mathbf{k}_i, \mathbf{v}_i \in \mathbb{R}^{1 \times d}$ , correspond to the  $i$ -th token's query, key, and value in traditional attention. The softmax function  $\frac{\exp(\mathbf{q}_i \mathbf{k}_j^T)}{\sum_{j=1}^N \exp(\mathbf{q}_i \mathbf{k}_j^T)}$  can be viewed as a weighting coefficient based on the similarity between  $x_i$  and  $x_j$ . (Katharopoulos et al., 2020) treat this similarity calculation method as one of the general functions  $\text{sim}(\cdot, \cdot)$  representing the interactions between different tokens. Linear attention uses a kernel function  $\mathcal{K}$  to represent the  $\text{sim}(\cdot, \cdot)$ , i.e.  $\text{sim}(\mathbf{q}_i, \mathbf{k}_j) := \mathcal{K}(q_i, k_j) = \sigma(\mathbf{q}_i) \sigma(\mathbf{k}_j^T)$ , here  $\sigma : \mathbb{R}^{1 \times d} \rightarrow \mathbb{R}^{1 \times d'}$  is a non-linear and positive map (Tiezzi et al., 2024; Tsai et al., 2019). Then the output can be written by the

following formula:

$$\mathbf{y}_i = \frac{\sum_{j=1}^N \text{sim}(\mathbf{q}_i, \mathbf{k}_j) \mathbf{v}_j}{\sum_{j=1}^N \text{sim}(\mathbf{q}_i, \mathbf{k}_j)} = \frac{\sum_{j=1}^N \sigma(\mathbf{q}_i) \sigma(\mathbf{k}_j^T) \mathbf{v}_j}{\sum_{j=1}^N \sigma(\mathbf{q}_i) \sigma(\mathbf{k}_j^T)}.$$

The function  $\sigma$  in linear attention serves to replace the traditional attention mechanism based on softmax's similarity. The splitting of the  $\text{sim}(\cdot, \cdot)$  allows the calculation order of  $\mathbf{Q}$ ,  $\mathbf{K}$ , and  $\mathbf{V}$  to be swapped so that the complexity of the calculation does not need to increase with the quadratic complexity of the sequence length. For details, please refer to (Katharopoulos et al., 2020).

Our work generalizes this computation method from vectors to matrices and realizes information aggregation among different texts. Combined with the equations 3 and 7,  $\mathbf{A}_{\text{ret}}$  can be represented by the following formula:

$$\begin{aligned} \mathbf{A}_{\text{ret}} &= \frac{\sigma(\mathbf{Q}) \mathbf{M}_k}{\sigma(\mathbf{Q}) \mathbf{n}_k} = \frac{\sigma(\mathbf{Q}) \sum_{i=1}^k \sigma(\mathbf{K}^{d_i})^T \mathbf{V}^{d_i}}{\sigma(\mathbf{Q}) \mathbf{n}_k} \\ &= \frac{\sum_{i=1}^k \sigma(\mathbf{Q}) \sigma(\mathbf{K}^{d_i})^T \mathbf{V}^{d_i}}{\sigma(\mathbf{Q}) \mathbf{n}_k}. \end{aligned}$$

Here, it can be considered that  $\sigma(\mathbf{Q}) \sigma(\mathbf{K}^{d_i})^T$  is the approximation of the  $\text{sim}(\cdot, \cdot)$  function acting on the matrix, representing the "similarity" between the query  $\mathbf{Q}$  and each demonstration  $d_i$ . Besides, knowing that our approach involves giving each existing demonstration  $d_i$  interaction with the query

Model	RAMS	RAMS	WIKI	WIKI	ACE05	ACE05	Avg
	↓ WIKI	↓ ACE05	↓ RAMS	↓ ACE05	↓ RAMS	↓ WIKI	
PAIE	20.5	32.4	32.2	48.5	20.3	40.6	32.4
PAIE-CMR (Ours)	<b>26.8</b>	<b>35.1</b>	<b>34.9</b>	<b>51.1</b>	<b>23.8</b>	<b>45.8</b>	<b>36.3</b>

Table 8: Performance metrics (Arg-C F1 score) across various  $\text{src} \Rightarrow \text{tgt}$  configurations are detailed. The model is trained on the  $\text{src}$  dataset and evaluated on the  $\text{tgt}$  dataset. The Avg column reflects the mean scores from all  $\text{src} \Rightarrow \text{tgt}$  scenarios.

Method	#N	#Demo BS	RAMS
			Inference Time (s)
PAIE	0	-	22.95
	1	1	46.18
	5	1	136.21
PAIE-CMR	5	4	90.72
	10	1	227.26
	10	4	141.75
	15	1	356.44
	15	4	206.32

Table 9: Inference time (second) for PAIE and PAIE-CMR on the test set of RAMS dataset. Experiments are run on one same RTX 4090 GPU. # Demo BS denotes the batch size of processing demonstrations.

$\mathbf{Q}$ , closely related to the demonstration  $d_i$  itself, this whole process can be understood through a selection mechanism (Gu and Dao, 2023): retaining important information among  $\{d_1, d_2, \dots, d_k\}$  related to the query  $\mathbf{Q}$  and discarding unimportant information. A function  $f(\mathbf{Q}, d_i) = \sigma(\mathbf{Q})\sigma(\mathbf{K}^{d_i})^T$  determines the importance of the demonstration  $d_i$ , influences how the representation  $\mathbf{V}^{d_i}$  acts on the final representation of input, i.e.  $\mathbf{A}_{\text{ret}}$ . Therefore, this process can be viewed as the query  $\mathbf{Q}$  retrieving information from the candidate demonstrations.

## D Efficiency Analysis

In this section, we explore the efficiency of the CMR mechanism. We compare the inference time of PAIE-CMR and PAIE on the RAMS test set. For PAIE-CMR, we measure the time required to retrieve 1, 5, 10, and 15 demonstrations. The inference batch size is set to 1, and the demonstration batch size  $B_j$  is 4.

As shown in Table 9, our PAIE-CMR model increases inference time compared to PAIE due to the need to store demonstrations. However, this additional time is justified by the corresponding improvement in performance. Moreover, by processing demonstrations in batches, our approach effectively reduces the overall time cost during in-

ference.

## E Demonstration Diversity Analysis

In this section, we analyze the improvement in diversity when retrieving multiple demonstrations compared to retrieving only the top 1 demonstration. We provide a specific case to illustrate this. As shown in Figure 6, the example case is an instance randomly selected from the RAMS dataset. Below are the demonstrations retrieved using SBERT based on similarity. It is evident that retrieving the top 5 demonstrations, compared to just the top 1, results in a greater diversity of event types. A more diverse set of demonstrations can provide richer retrieval information, ensuring the effectiveness of RAG.

<p><b>Example Case:</b>  <b>Context:</b> Like her role in handing Russia exclusive mining rights to 20 percent of US uranium reserves via a company that donated millions to the foundation . You thought Donald Trump was Vladimir Putin 's best friend ? Or the tens of millions donated by the same Middle Eastern...  <b>Target Event Type:</b> transaction.transfermoney.giftgrantprovideaid</p>
<p><b>Top1 [Similarity: 0.80367112159729]</b>  <b>Context:</b> The newly disclosed email, part of the thousands of hacked documents being made public by WikiLeaks, gives a sense of the scale of that giving from one U.S. ally with, at times, complicated relations with the American government...2012, apparently so that he could present a \$1 million check that Qatar had promised in honor of Clinton's birthday in 2011.  <b>Event Type:</b> transaction.transferownership.n/a</p>
<p><b>Top2 [Similarity: 0.7551537156105042]</b>  <b>Context:</b> How is it that Clinton still claims that she is "THE MOST TRANSPARENT CANDIDATE IN MODERN TIMES" when her Foundation Failed to Disclose 1,100 Foreign Donations. And this is on top of Clinton's Failure once again to disclose more than \$26,000,000 in speaking fees "from foreign governments and corporations" Because you know ...  <b>Event Type:</b> transaction.transaction.giftgrantprovideaid</p>
<p><b>Top3 [Similarity: 0.7524813413619995]</b>  <b>Context:</b> The investigation by the Daily Caller News Foundation has uncovered a disturbing pattern of the Clintons raising money for the Clinton Foundation from regimes that have checkered records on human rights and that aren't always operating in the best interests of...  <b>Event Type:</b> transaction.transfermoney.purchase</p>
<p><b>Top4 [Similarity: 0.7116298675537109]</b>  <b>Context:</b> And in 2013, nearly 100 percent of the \$4.4 million of the government donations came from overseas governments. Only \$23,000 came from U.S. government entities, according to the exhibit. The disclosures likely will fuel charges by presumptive Republican presidential candidate Donald Trump, who claims Clinton turned her secretaryship into a huge  <b>Event Type:</b> contact.commandorder.n/a</p>
<p><b>Top5 [Similarity: 0.7020836472511292]</b>  <b>Context:</b> Bill and Hillary Clinton epitomize the use of public office in behalf of the office holder's interest. For the Clintons, government means using public office to be rewarded for doing favors for private interests. The Wall Street Journal reported that "at least 60 companies that lobbied the State Department during her [Hillary Clinton's] tenure as Secretary of State donated a total of more than \$26 million to the Clinton Foundation...  <b>Event Type:</b> transaction.transferownership.giftgrantprovideaid</p>

Figure 6: A specific case from the RAMS dataset highlighting the importance of diversity in demonstrations.