GEMS: Generation-Based Event Argument Extraction via Multi-perspective Prompts and Ontology Steering

Run Lin, Yao Liu*, Yanglei Gan, Yuxiang Cai, Tian Lan, Qiao Liu
University of Electronic Science and Technology of China
{runlin, yangleigan, yuxiangcai}@std.uestc.edu.cn,
{liuyao, lantian1029, qliu}@uestc.edu.cn

Abstract

Generative methods significantly advance event argument extraction by probabilistically generating event argument sequences in a structured format. However, existing approaches primarily rely on a single prompt to generate event arguments in a fixed, predetermined order. Such a rigid approach overlooks the complex structural and dynamic interdependencies among event arguments. In this work, we present GEMS, a multi-prompt learning framework that **G**enerates **E**vent arguments via **M**ulti-perspective prompts and ontology **S**teering. Specifically, GEMS utilizes multiple unfilled prompts for each sentence, predicting event arguments in varying sequences to explicitly capture the interrelationships between arguments. These predictions are subsequently aggregated using a voting mechanism. Furthermore, an ontology-driven steering mechanism is proposed to ensure that the generated arguments are contextually appropriate and consistent with event-specific knowledge. Extensive experiments on two benchmark datasets demonstrate that GEMS achieves state-of-the-art performance, particularly in low-resource settings. The source code is available at: https://github.com/AONE-NLP/EAE-GEMS.

1 Introduction

Event argument extraction (EAE) is a crucial yet challenging task in Natural Language Understanding (NLU), focused on identifying role-specific spans of text within a given event (Sundheim, 1992; Chen et al., 2015; Sha et al., 2018; Du and Cardie, 2020a). For instance, consider the sentence from the ACE05 dataset, ***Pearl*** *was* *murdered* *by* ***terrorists*** *in* ***Pakistan***. The verb "murdered" triggers a "*Life.Die*" event, and the EAE task aims to identify "Pearl", "terrorists", and "Pakistan" as event arguments with the roles of "victim", "agent", and "place", respectively.

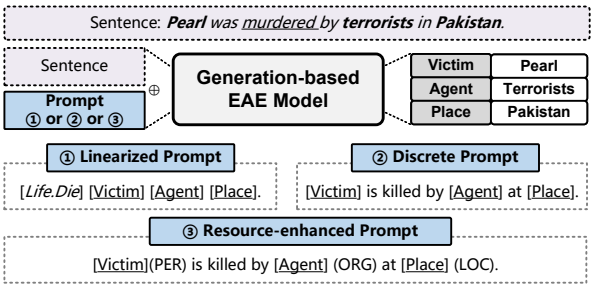


Figure 1: Example of different prompt designs that various generation-based EAE model utilize.

Traditional approaches to EAE generally involve first identifying potential argument candidates and then assigning specific roles via multi-label classification (Wadden et al., 2019; Lin et al., 2020; Liu et al., 2024). Other methods treat EAE as a reading comprehension problem (Du and Cardie, 2020b; Liu et al., 2020), employing question templates to guide the extraction process. In contrast, recent advancements in generation-based EAE methods (Li et al., 2021; Lu et al., 2021; Hsu et al., 2022; Ma et al., 2022) have demonstrated superior generalizability and competitive performance. These methods offer enhanced flexibility (Liu et al., 2021), as they can easily accommodate new event types with minimal adjustments to the prompts and decoding schemes.

In this context, generation-based methods probabilistically populate the structured slots by leveraging contextual semantics. Various prompt designs, shown in Figure 1, are employed to guide this process: (1) *linearized prompts* (Lu et al., 2021, 2022; Wang et al., 2023; Ren et al., 2023), where arguments are presented in a sequentially ordered format, following a predefined structure; (2) *discrete prompts* (Li et al., 2021; Ma et al., 2022; Hsu et al., 2023a; Luo and Xu, 2023; Zhang et al., 2024; Li et al., 2024), which integrate natural lan-

*Corresponding Author

Findings of the Association for Computational Linguistics: ACL 2025, pages 26392–26409

guage with special tokens to designate roles or labels; (3) *resource-enhanced prompts* (Hsu et al., 2023b; Zhang et al., 2023; Wang et al., 2023; Yang et al., 2024), which incorporate external knowledge (e.g., syntactic information), to guide the generation by providing additional contextual cues.

Despite the effectiveness of these methods, they typically rely on fixed left-to-right argument generation orders, which overlook the complex interdependencies among event arguments, leading to potential limitations: (1) **Insufficient capture of argument correlations**. Unlike natural text generation, prompt-based argument generation struggles with fixed-sequence structures, which are incompatible with the dynamic and interdependent nature of event arguments (Li et al., 2023); (2) **Ineffective knowledge querying**. Existing methods struggle to query relevant information effectively during decoding (Bouraoui et al., 2020; Schick and Schütze, 2021a). As a result, the generated outputs often reflect only a lower bound of the model's knowledge (Jiang et al., 2020a), leading to arguments that may be contextually inconsistent or semantically inaccurate; (3) **Inconsistent performance**. Fixed-order generation approach struggles to maintain uniformity across different event contexts (Hu et al., 2022; Liu et al., 2021). This inconsistency stems from the rigidity of prompt design, which fails to adapt to the diverse structure and interdependency of event arguments, leading to fluctuations in performance (Liu et al., 2022).

In this paper, we investigate generation-based EAE within a multi-prompt learning scheme and introduce a novel approach GEMS, that systematically **G**enerate **E**vent arguments via **M**ulti-perspective prompts and ontology-guided **S**teering. Specifically, we propose a strategy that employs multiple unanswered prompts for a given sentence, each predicting event arguments in different orders to explicitly capture the interactions among multiple arguments (Sec 3.2). The predictions from these diverse prompts are then ensembled, exploiting the fact that different prompts may be more effective for querying context oriented towards specific arguments. To ensure coherent generation, we integrate an ontology-driven steering mechanism via a dual-branch cross-attention process (Sec 3.3). This guides the decoding process, ensuring that the generated event arguments are both contextually appropriate and aligned with the event-specific knowledge defined in the ontology. Our main contributions are as follows:

- We approach the challenge of event argument extraction (EAE) by explicitly modeling the interactions among event elements through multivariate prompt permutation, effectively addressing the complex dependencies between arguments.
- We incorporate an ontology-guided steering mechanism through a dual-branch cross-attention process, ensuring the generation are not only contextually relevant but also consistent with the event-specific knowledge defined in the ontology.
- Extensive experiments on `ACE05-E` and `ERE-EN` datasets show that GEMS outperforms state-of-the-art models in EAE task, particularly in low-resource scenarios.

2 Related Work

In recent years, significant progress has been made in the task of Event Argument Extraction (EAE). Early approaches predominantly relied on **multi-label classification methods**, which focus on identifying argument spans and assigning them corresponding role labels. These approaches often incorporate auxiliary syntactic structures (Liu et al., 2018; Pouran Ben Veyseh et al., 2020; Zhang and Ji, 2021) or model semantic relationships among event elements through sophisticated network architectures (Chen et al., 2015; Sha et al., 2018; Wadden et al., 2019; Lin et al., 2020; Ding et al., 2022; Xu et al., 2023; Liu et al., 2024). While these methods have improved performance, they typically lose focus on the broader context once candidate argument spans are identified, leading to a loss of contextual semantic integrity.

To address these limitations, **sequential generation methods** focus on leveraging complete contextual information throughout the extraction process to guide the model's decoding. These methods emphasize carefully designed prompts that direct the generation of sequences centered around arguments and their roles, transforming the argument extraction task into a sequence-to-sequence generation problem. Early research employed **linearized prompts** (Paolini et al., 2021; Ren et al., 2023), which directly concatenate arguments with their corresponding roles in a target format. To better capture the event-based organizational structure of argument roles, Li et al. (2023) incorporated event structures to establish role dependencies, combining event roles with soft prompts. Ad-

ditionally, Lu et al. (2021, 2022); Wang et al. (2023) constructed event elements in a linearized target format using a structured templates.

More recent advancements have focused on incorporating richer representations of argument roles and event structures by leveraging **discrete prompts** (Li et al., 2021; Hsu et al., 2023a; Zhang et al., 2024). These prompts involve manually designed, human-introspected templates tailored to each event type, which enhance the models ability to capture the nuanced semantic relationships among event roles (Ma et al., 2022; Hsu et al., 2022; Luo and Xu, 2023; Li et al., 2024). Simultaneously, several studies have introduced **resource-enhanced prompts**, drawing on external auxiliary resources such as syntactics and abstract meaning representation (AMR) structures (Hsu et al., 2023b; Zhang et al., 2023; Wang et al., 2023; Yang et al., 2024). These resource-enhanced approaches further enrich the context and event schemas used during extraction, improving the quality of argument role representation. Despite the improvements in performance, these methods reliance on rigid, predefined prompts limits their flexibility in adapting to the interdependencies of event arguments, leading to fluctuations in performance.

To overcome these limitations, recently, **prompt ensembling**, a method of using multiple prompts during inference, has been shown to reduce performance variations and enhance the generalization of language models on downstream tasks (Liu et al., 2023). It leverages the complementary advantages of different prompts to boost prediction stability and consistency across various linguistic and semantic contexts (Lee et al., 2025; Tonolini et al., 2024). Inspired by traditional ensemble learning in machine learning and deep learning (Zhou et al., 2002), prompt ensembling varies in how it aggregates predictions from different prompts, with methods ranging from simple averaging of predicted probabilities or logits (Jiang et al., 2020b; Schick and Schütze, 2021a) to more sophisticated mechanisms like weighted averaging (Qin and Eisner, 2021; Schick and Schütze, 2021b; Khattak et al., 2023) and majority voting (Lester et al., 2021).

3 Methodology

3.1 Task Definition

We formulate EAE task as a prompt-based argument generation problem defined on a given context C. Let $S = \{s_1, s_2, ..., s_{|x|}\}$ denote the token sequence of C, and e be a predetermined event type with $t \in S$ as corresponding trigger. Given the event-specific role set R^e, the objective of the EAE is to extract argument spans $a_i \in S$ and assign each argument a_i a corresponding role $e_i \in R^e$, resulting in argument-role tuples $\{a_i^e, r_i^e\}$.

3.2 Multi-perspective Prompt Design

We introduce a multi-perspective prompting mechanism to systematically control the prediction order of event elements. To this end, we employ dedicated element markers that encode the structural roles of event components (Paolini et al., 2021). Specifically, we use [T] for trigger terms t^e, and [A] together with [R] for arguments a_i^e and their associated roles r_i^e in the set $\{(a_i^e, r_i^e)|1 \leq i \leq |R^e|\}$. Each element is prefixed with its corresponding marker, while a special symbol [SSEP] is employed to concatenate these components in a specified permutation p_i.

Element-wise Prompt Permutation. To explore diverse structural configurations of trigger terms, arguments, and corresponding roles, we define four distinct element-wise prompt permutations: [T] [A] [R]; [T] [R] [A]; [A] [R] [T]; [R] [A] [T][1]. [T] [A] [R] indicates that both the prompt construction and the subsequent prediction should follow the order $t \Rightarrow a \Rightarrow r$.

To illustrate, consider the sentence: *Police arrests a killer.*, with the event type "*Justice.Arrest-Jail*" and three pre-defined argument roles: *agent*, *person* and *place*. For the permutation [T] [A] [R], we concatenate S with prompt to obtain x^e and the respective input-target sequence pair $(x_{p_i}^e, g_{p_i}^e)$ for training is constructed as follows:

Input $(x_{p_i}^e)$**:** *Police arrests a killer.* [T] *arrests* [SSEP] [A] [R] *place* [SSEP] [A] [R] *person* [SSEP] [A] [R] *agent*.

Target $(g_{p_i}^e)$**:** [T] *arrests* [SSEP] [A] *null* [R] *place* [SSEP] [A] *killer* [R] *person* [SSEP] [A] *Police* [R] *agent*.

Argument-wise Prompt Permutation. In addition to the element-wise permutation, we introduce an argument-wise prompt permutation scheme to explicitly capture interactions among multiple arguments within a single event. Under this scheme, the arguments of an event "Justice.Arrest-Jail" can be rearranged in various valid orders (e.g., *agent*, *person*, *place*; *place*, *per-*

[1]We do not permute the trigger marker [T] between [A] and [R], as interleaving it would exponentially increase prompt configurations, complicating model training.

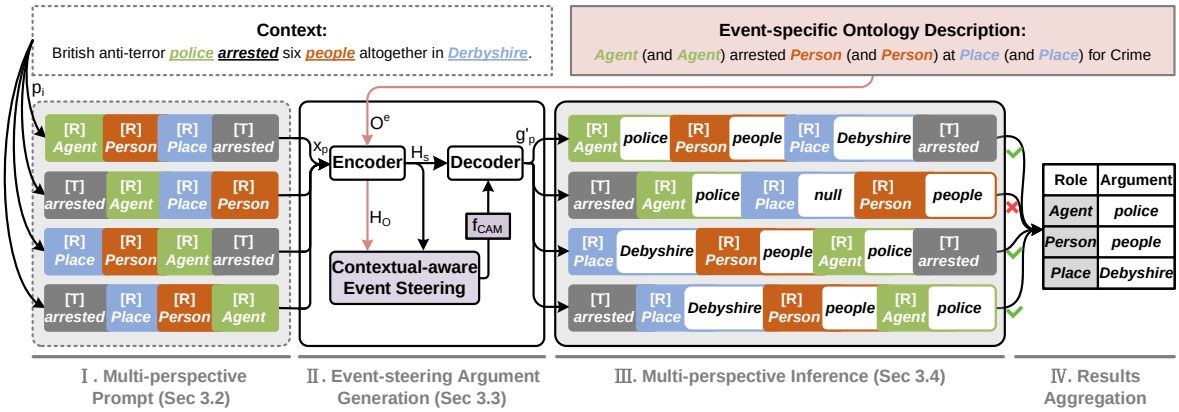


Figure 2: Overview of GEMS structure. GEMS generates multiple prompts by permuting event elements and encoding them with concatenated context and event-specific ontology. The decoder then generates argument sequences for each permutation independently, capturing diverse structural interactions among event arguments. The predictions are then aggregated through majority voting for final results.

son, *agent*; *person*, *place*, *agent*), reflecting permutations of different arguments in a single event. This flexibility allows our model to account for varying argument interactions. When multiple arguments share the same role, we simply concatenate their terms in the order in which they appear.

3.3 Event-steering Argument Generation

Given contextual sequence S with m randomly selected prompt permutations and event-specific ontology O, this module generates event arguments by leveraging event-specific context, utilizing standard Transformer-based Encoder and Decoder.

Encoder. We first leverage a multi-layer transformer encoder to obtain the semantic representations for contextual token sequence with prompt $x^e_{p_i}$ and event-specific ontology description $\mathbf{O^e}$:

$$\begin{aligned} \mathbf{H}_s &= Encoder(x^e_{p_i}) \in \mathbb{R}^{n_s \times d}; \\ \mathbf{H_O} &= Encoder(\mathbf{O}^e) \in \mathbb{R}^{n_\mathbf{O} \times d}, \end{aligned} \tag{1}$$

where n_s and $n_\mathbf{O}$ are the maximum token lengths of the input text and the event-specific ontology description, respectively, and d denotes the embedding dimension.

Contextual-aware Event Steering. To guide the generation distribution with knowledge from the event ontology, we adopt a translation transformation factor f_{CAM} following insights from Han et al. (2024). This factor is designed to steer each word embedding during decoding toward the target semantic space defined by the event ontology. Concretely, f_{CAM} is derived via a dual-branch cross-attention mechanism (CAM) that captures contextual associations between the contextual sequence $\mathbf{H}_s$ and the event-specific ontology $\mathbf{H}_O$.

We then project $\mathbf{H}_s$ and $\mathbf{H}_O$ into two distinct semantic spaces, context-oriented space and ontology-oriented space, respectively, as follows:

$$h^s = \mathbf{H}_s^\top W^s \in \mathbb{R}^{1\times d}; \quad h^\mathbf{O} = \mathbf{H}_\mathbf{O}^\top W^\mathbf{O} \in \mathbb{R}^{1\times d}, \tag{2}$$

where $W^s \in \mathbb{R}^{n_s \times 1}$, $W^\mathbf{O} \in \mathbb{R}^{n_\mathbf{O} \times 1}$ are trainable matrices. To integrate the contextual features from both semantic spaces, we treat h^s and $h^\mathbf{O}$ as CLS-like tokens, enabling them to exchange semantics with word tokens in the other semantic space ($\mathbf{H_O}$ and $\mathbf{H}_s$) via CAM. Formally, the attention operations are defined as:

$$\begin{aligned} q^s = W_q h^\mathbf{O}, k^s = W_k \left[\mathbf{H}_s \parallel h^\mathbf{O}\right], v^s = W_v \left[\mathbf{H}_s \parallel h^\mathbf{O}\right], \\ q^\mathbf{O} = W_q h^s, k^\mathbf{O} = W_k \left[\mathbf{H_O} \parallel h^s\right], v^\mathbf{O} = W_v \left[\mathbf{H_O} \parallel h^s\right], \\ h'^s = softmax\left(q^s {k^s}^\top / D_n\right) v^s + h^s, \\ h'^\mathbf{O} = softmax\left(q^\mathbf{O} {k^\mathbf{O}}^\top / D_n\right) v^\mathbf{O} + h^\mathbf{O}, \end{aligned} \tag{3}$$

where $\parallel$ denotes concatenation operations. W_q, W_k, $W_v \in \mathbb{R}^{d \times {D_n}^2}$ are learnable parameters. a is the number of attention heads. D_n denotes as $\sqrt{d/a}$. Finally, the output factor f_{CAM} of the CAM module, which integrates contextual association information h'^s and $h'^\mathbf{O}$ for both semantic spaces via layer normalization is formulated as:

$$f_{CAM} = LN(h'^s) + LN(h'^\mathbf{O}). \tag{4}$$

Decoder. We condition on text with prompt x to generate the output g', modeled as:

$$g'_j, h_{g'_j} = Decoder(\mathbf{H}_s, g'_{j-1}, f_{CAM}), \tag{5}$$

During decoding, we redesign the calculation of each word output likelihoods. Typically, From the

view of word embedding, the dot product $H_s^\top h_{g'_j}$ between a computed context vector and a learnable output word embedding $h_{g'_j}$ for token g'_j is usually used as the word logit. In our model, we additionally apply a linear factor f_{CAM} from attention-aware event steer and the final token probability P among whole vocabulary $\mathcal{V}$ is defined as follows:

$$\mathbf{P}(g'_j|\mathbf{H}_s, f_{CAM}) = \frac{\exp\left(\mathbf{H}_s^\top\left(h_{g'_j} + f_{CAM}\right)\right)}{\sum_{u\in\mathcal{V}}\exp\left(\mathbf{H}_s^\top h_u\right)}, \quad (6)$$

Training. From the view of overall model training, given the input-target pair (x, g) as described in Section 3.2. we can fine-tune a pre-trained sequence-to-sequence language model, minimizing the following negative log-likelihood loss:

$$\begin{aligned}\mathcal{L}_{NLL} &= -\mathbb{E}\log \mathrm{L}\left(g|x\right) \\ &= -\mathbb{E}\sum_{j=1}^{n_g}\log \mathrm{L}\left(g'_j|x, g'_{<j}\right)\end{aligned} \quad (7)$$

where n_g is the length of the target sequence g and $g'_{<j}$ denotes previously generated tokens.

3.4 Multi-perspective Inference

During inference, we prompt the trained model with randomly selected m prompt permutations. Each permutation p_i produces a set of predicted tuples D_{p_i}, where each set may contain one or more argument-role pairs. We then aggregate these sets and retain the most frequently appearing tuples across perspectives as our final prediction. Formally, the aggregated result D_{agg} is defined as:

$$D_{agg} = \{k|k \in \bigcup_{i=1}^{m} D_{p_i} \text{ and } (\sum_{i=1}^{m}\mathbb{1}_{D_{p_i}}(k) \geq \frac{m}{2})\} \quad (8)$$

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate our model on two widely-used EAE benchmarks: ACE05-E[3] and ERE-EN[4]. Following prior work (Wadden et al., 2019; Lin et al., 2020; Hsu et al., 2022, 2023b), we preprocess each dataset by splitting documents into individual sentences. Furthermore, we follow Hsu et al. (2023b) and create different training splits by sampling 5%, 10%, 20%, 30%, and 50% of the original training data. See Appendix B.2 for detailed statistics of datasets.

[3]https://catalog.ldc.upenn.edu/LDC2006T06

[4]https://catalog.ldc.upenn.edu/LDC2020T19

Evaluation Metrics. Following prior studies (Lin et al., 2020; Ren et al., 2023; Hsu et al., 2023b), we report F1-scores for argument prediction: (1) Argument Identification F1 score (**Arg-I**): An argument is considered correctly identified if its predicted span exactly matches the span of any gold-standard argument; (2) Argument Classification F1 score (**Arg-C**): An argument is considered correctly classified if the predicted span and the role type both match the gold reference.

Baselines. To evaluate the performance of GEMS, we compare it with the following models: (1) Multi-label Classification methods: **DyGIE++** (Wadden et al., 2019), **OneIE** (Lin et al., 2020), **Query and Extract** (Wang et al., 2022), **AMR-IE** (Zhang and Ji, 2021), and **DEEIA** (Liu et al., 2024); (2) Sequential Generation methods: **PAIE** (Ma et al., 2022), **DEGREE** (Hsu et al., 2022), **AMPERE** (Hsu et al., 2023b), **TagPrime** (Hsu et al., 2023a), and **Scented-EAE** (Yang et al., 2024). We provide baseline descriptions and implementation details of GEMS in Appendix B.1.

4.2 Overall Performance

Table 1 presents a performance comparison between our proposed GEMS and state-of-the-art (SOTA) baselines on both ACE05-E and ERE-EN datasets under varying training data proportions (experimental results for argument identification are listed in Appendix C.4). Our approach consistently achieves superior results across all datasets, demonstrating robustness regardless of the proportion of training data utilized. Further analysis of the experimental results reveals that: (1) **GEMS outperforms other generation-based EAE models** that utilize a single fixed prompt for each event type, indicating that the multi-perspective prompt design effectively leverages the complementary strengths of various prompt permutations. (2) **The performance gap between GEMS and SOTA models widens as the training data size decreases**. Under a fully supervised setting, the gap between GEMS and the second-best model is 0.6% on ACE05-E and 1.1% on ERE-EN. However, as the training data proportion reduces to just 5%, the gap increases significantly to 2.4% on ACE05-E and 3.3% on ERE-EN. Unlike other models, GEMS leverages its multi-perspective prompt design to guide the extraction of arguments, effectively compensating for the lack of large-scale supervision. (3) **Multi-label classification models underperform compared to sequential generation mod-**

Model	ACE05-E Test Set						ERE-EN Test Set					
	5%	10%	20%	30%	50%	100%	5%	10%	20%	30%	50%	100%
DyGIE++ (2019)	29.3	42.2	49.5	53.2	54.4	57.4	40.0	44.6	49.5	52.0	53.7	56.0
OneIE (2020)	34.6	50.0	59.6	63.0	68.4	70.7	49.5	56.1	62.3	66.1	67.7	70.1
AMR-IE (2021)	36.8	48.5	58.3	62.6	66.1	70.3	44.1	53.7	60.4	65.7	68.9	71.5
Query and Extract (2022)	11.0	20.9	34.3	44.3	49.6	59.1	19.7	34.0	42.4	50.1	57.7	64.3
DEEIA (2024)♣	38.4	50.5	55.7	65.4	69.1	72.3	58.3	60.0	63.0	66.3	68.7	72.0
PAIE (2022)	46.3	56.3	62.8	65.8	69.1	72.1	57.4	64.4	64.6	68.3	69.1	73.1
DEGREE (2022)	41.7	57.7	58.9	65.8	68.2	73.0	57.5	63.9	67.4	69.1	73.3	74.9
AMPERE$_{\text{AMRBart}}$ (2023b)♣	50.7	58.9	66.2	68.5	70.7	72.3	63.3	66.3	68.0	70.9	71.4	74.5
AMPERE$_{\text{AMRRoberta}}$ (2023b)♣	51.5	58.7	64.7	68.6	71.0	72.5	63.9	67.4	67.8	70.0	71.7	73.7
TagPrime (2023a)♣	43.2	56.8	66.7	69.8	72.9	73.8	64.5	66.4	69.5	73.1	73.3	75.7
Scented-EAE (2024)♣	50.2	60.1	66.8	69.0	70.9	73.1	61.3	65.4	68.2	71.0	72.0	73.2
GEMS	**53.9**	**61.7**	**67.9**	**69.9**	**73.0**	**74.6**	**67.2**	**69.1**	**70.4**	**73.4**	**74.1**	**76.8**

Table 1: Argument classification F1-scores (%) under different training data proportion settings for ACE05-E and ERE-EN datasets. The best F1-scores are denoted in **bold** and the second highest scores are underlined. We re-implemented the methods marked with ♣ by using their released code[2], running each model *five times* with different random seeds, and report the **average F1-scores**. The rest are retrieved from (Hsu et al., 2023b).

Settings		ACE05-E			ERE-EN		
		50%	30%	10%	50%	30%	10%
p_i	[T] [A] [R]	72.4	68.2	60.2	74.5	73.2	69.0
	[T] [R] [A]	73.5	68.5	60.0	73.6	72.9	68.8
	[A] [R] [T]	72.4	69.6	59.8	74.0	72.8	68.1
	[R] [A] [T]	72.4	70.3	61.3	74.0	72.8	68.5
Agg	GEMS$_{random}$	72.2	68.5	60.6	74.0	72.8	68.2
	GEMS$_{rank}$	72.5	69.7	60.8	73.9	72.6	68.2
	GEMS$_{vote}$	**73.0**	**69.9**	**61.7**	**74.1**	**73.4**	**69.1**

Table 2: Performance on different element-wise prompt permutation and different aggregation strategies for Arg-C. The best results are marked in **bold**.

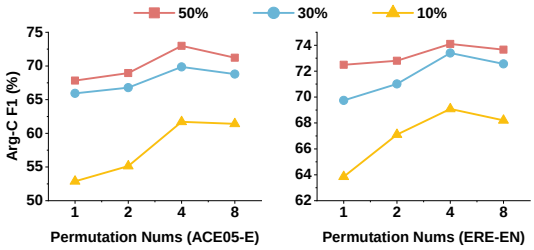


Figure 3: Performance on various numbers of prompt permutations under different training data proportions.

els. This performance gap arises from the challenges multi-label models face in capturing complex dependencies and correlations between event arguments. In contrast, sequential generation models, particularly GEMS, excel in both datasets, with GEMS demonstrating a pronounced advantage in data-scarce settings.

4.3 Ablation Study

To illustrate the effectiveness of our proposed multi-perspective prompt and event-specific steering modules in GEMS, we conduct ablation studies on ACE05-E and ERE-EN datasets in Table 2.

Sensitivity to different prompt permutations. As shown in the top portion of Table 2, we analyze the performance achieved under several prompt permutations. Our findings demonstrate that no single permutation consistently outperforms the others across varying training data sizes, highlighting the importance of aggregating outputs from multiple, diverse prompts. This observation motivates the exploration of ensemble-like strategies to harness the complementary strengths of each permutation. Full ablation studies under various training data ratios can be found in Appendix C.1.

Impact of different aggregation strategies. We further explore three aggregation strategies for generation results of our multi-perspective prompts. Specifically, GEMS$_{rank}$ picks the top-ranked output sequence according to a prediction score according to Equation (9), while GEMS$_{random}$ randomly samples one. In contrast, GEMS$_{vote}$, which employs majority voting, achieves the most stable and robust results, with F1 scores of 73.0%, 69.9%, and 61.7% on ACE05-E and 74.1%, 73.4%, and 69.1% on ERE-EN under varying training data sizes. The 0.8% to 1.1% absolute F1 improvement over the other aggregation strategies in the 10% data setting underscores the importance of leveraging multiple perspectives to mitigate instability in low-resource conditions.

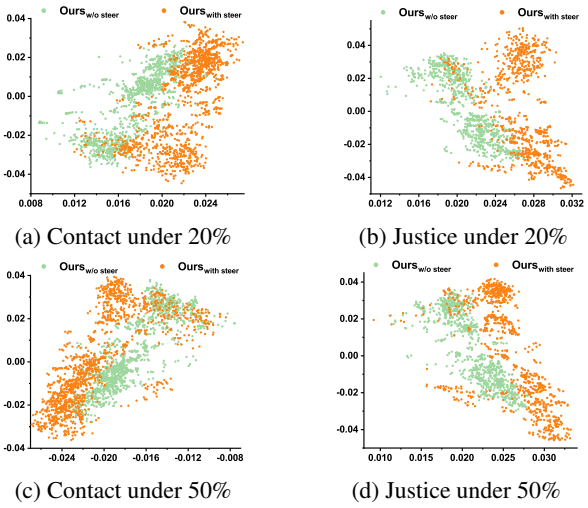
[4]We design manual templates for PAIE based on the ERE-EN dataset to obtain the corresponding results.

Settings	50%		20%	
	Arg-I	Arg-C	Arg-I	Arg-C
GEMS	**76.3**	**73.0**	**72.0**	**67.9**
w/o h'^{o}	73.8	70.2	70.0	64.7
w/o h'^{s}	73.1	69.6	69.3	64.5
w/o steer	69.0	65.1	63.4	59.4

Table 3: Effect of different event steer settings in ACE05-E dataset (*F1-score,%*).

Settings	50%		20%	
	Std.↓	*BBox Vol.*↑	*Std.*↓	*BBox Vol.*↑
$\text{GEMS}_{\text{with steer}}$	**0.076**	**647.0**	**0.070**	**586.0**
$\text{GEMS}_{\text{w/o steer}}$	0.081	524.8	0.077	457.1

Table 4: Results of standard deviation (*Std.*) and log bounding box volume (*BBox Vol.*) of output argument word representations from decoder in ACE05-E dataset. ↓ means lower is better, while ↑ means higher is better. The best results are denoted in **bold**.



(a) Contact under 20% (b) Justice under 20%

(c) Contact under 50% (d) Justice under 50%

Figure 4: Visualization of output argument word representations during decoding. We compare the dispersion area in samples associated with two event types (Contact, Justice) under two model settings (with and without Event Steer) in ACE05-E dataset.

Impact of various numbers of permutations. We investigate how the GEMS's performance varies under different numbers of prompt permutations (m). As illustrated in Figure 3, transitioning from single-perspective to multi-perspective training and inference yields substantial performance gains. Specifically, the Arg-C F1-score improves by 1.61% to 8.83% as the number of permutations increases from 1 to 4, with the most significant improvements observed under 10% training data. Beyond 4 permutations, performance begins to plateau and slightly decline within the 4 to 8 range, suggesting diminishing returns and potential noise introduced by excessive prompt permutations. The optimal configuration consistently falls with 4 prompt permutations in our case.

4.4 Effectiveness of Event-specific Steering

Table 3 presents the F1-score results of GEMS under different event-specific steering settings on ACE05-E, evaluated with 50% and 20% of the training data. The removal of event-specific steering (w/o steering) results in substantial performance degradation, with F1 scores dropping by 7.3% and 8.6%. To further analyze the contributions of different steering components, we evaluate the effect of removing each branch of the cross-attention module. The exclusion of ontology-oriented steering (w/o h'^{O}) leads to 2.0% to 3.2% performance drop, indicating that structured knowledge from the ontology space is essential for guiding event argument extraction. Similarly, removing context-oriented semantic representation (w/o h'^{s}) leads to a similar 2.7% to 3.4% performance decline. These effects are more pronounced with 20% training data. These findings confirm that event-specific steering is particularly beneficial in data-scarce environments, providing structured guidance that enhances model robustness.

To assess the impact of event steering on stabilizing decoding and enhancing token diversity, we calculate the standard deviation of cosine similarity between output argument word representations and the volume of their bounding boxes (Appendix C.2). As shown in Table 4, employing **Event-specific Steering** *significantly increases the log bounding box volume*, which reflects an expanded word representation space and promotes more diverse token generation. Additionally, we observe a reduction in the standard deviation of cosine similarity among tokens, suggesting a more uniform token distribution across generated arguments. These results demonstrate that event steering enhances the diversity of the representation space, improving the robustness of the model and mitigating the risk of representation collapse.

In order to more intuitively observe the guiding effect of event steering, we visualize the output argument word representations during the inference phase. Specifically, we collect the decoder output hidden states for samples of two event types (Contact, Justice) from the test dataset. Then, we apply Singular Value Decomposition (SVD) to re-

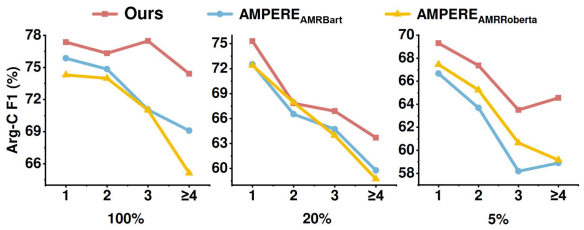


Figure 5: Performance on multiple events in ERE-EN.

Model	ACE05-E			ERE-EN		
	1%	2%	3%	1%	2%	3%
PAIE	31.1	35.8	42.6	16.5	44.3	49.1
AMPERE$_{AMRBart}$	34.5	33.8	45.5	34.7	52.6	58.9
AMPERE$_{AMRRoberta}$	36.2	37.1	46.7	32.6	53.4	58.5
TagPrime	25.7	30.3	41.6	33.6	49.8	57.3
Scented-EAE	28.1	30.2	45.4	32.2	45.3	58.2
DEEIA	24.2	31.2	39.5	26.4	41.0	47.6
GEMS	**36.8**	**40.5**	**48.5**	**36.0**	**54.9**	**59.8**

Table 5: Performance on extreme low-resource scenarios (*Arg-C F1,%*).

duce the hidden states of the argument tokens to a two-dimensional plane, as shown in Figure 4. After adding event steering, the dispersion area of tokens increases significantly, indicating that the GEMS's learned word vectors are more widely distributed in the feature space. This suggests better representation capability and enhanced generation diversity brought by the Event-specific Steering.

4.5 Performance on Multiple Events

In this section, we examine the effectiveness of the proposed method in handling multi-event scenarios. We utilize the ERE-EN dataset, which contains a higher proportion of multi-event instances compared to ACE05-E, and categorize the development sets based on event counts. As demonstrated in Figure 5, we observe a declining trend in performance for all models as the number of events increases across three different training data proportion scenarios. We attribute this decline to the increased complexity of processing more events, which requires the model to handle longer text and more intricate contextual relationships. Additionally, we observe a significant performance drop in AMPERE$_{AMRRoberta}$ and AMPERE$_{AMRBart}$ when the number of events exceeds two. In contrast, GEMS shows a notable improvement in multi-event contexts, highlighting its superiority in distinguishing the correlations between events.

Model	ACE05-E			
	1%	5%	10%	100%
UIE	12.8	30.4	36.3	69.3
InstructUIE$_{FlanT5-11B}$	-	-	-	56.8
GoLLIE$_{Code-LLaMA-34B}$	-	-	-	68.6
LLaMA$_{LLaMA2-7B}$	33.3	46.3	52.3	-
KnowCoder$_{LLaMA2-7B}$	**38.5**	48.3	55.1	70.3
GEMS$_{T5-large}$	36.8	**53.9**	**61.7**	**74.6**

Table 6: Comparison with fine-tuning LLMs (*Arg-C F1,%*).

4.6 Performance under Extreme Low-resource Settings

To further evaluate the effectiveness of GEMS in extreme low-resource scenarios, we follow the sampling strategy of Hsu et al. (2022) and train our model on 1%, 2%, and 3% of the training data, while testing on the full original test set. The results, presented in Table 5, demonstrate that GEMS consistently outperforms all baseline methods across both ACE05-E and ERE-EN datasets, reinforcing its robustness in data-scarce environments. Quantitatively, GEMS achieves 36.8, 40.5, and 48.5 Arg-C F1 on ACE05-E, surpassing the previous best-performing model, AMPERE$_{AMRRoberta}$, by 0.6%, 3.4%, and 1.8% points in the 1%, 2%, and 3% settings, respectively. Similar trend can be observed in ERE-EN dataset. Unlike other baselines that struggle with extreme data limitations, as PAIE achieves only 16.5% on the ERE-EN dataset with 1% data, GEMS maintains a relatively high 36.0%. These results underscore GEMS' advantage in low-resource settings, where its multi-perspective approach leverages diverse prompts and ontology-guided steering to extract event arguments.

4.7 Comparison with LLMs

To further assess the generalization ability of GEMS for EAE task, we compare its performance with fine-tuned Large Language Models (LLMs), following the experimental setup used in Li et al. (2024). We conduct this comparison across four different partitions of the original training sets (1%, 5%, 10%, 100%), and detailed descriptions of models are provided in Appendix B.1.

To evaluate the effectiveness of our design on LLMs, we replicate similar fine-tuning experiments using FlanT5-11B, Code-LLaMA-34B and LLaMA2-7B as described in Li et al. (2024). As

shown in Table 6, GEMS significantly outperforms the fine-tuned LLMs across all data splits. Specifically, GEMS achieves an Arg-C F1 score of 36.8% with 1% data, 53.9% with 5%, 61.7% with 10%, and 74.6% with 100%, outperforming the best performing LLM models at each respective data proportion, except with 1% training data. For instance, KnowCoder fine-tuned with LLaMA2-7B achieves 38.5% at 1%, 48.3% at 5%, and 55.1% at 10%, but GEMS shows a clear advantage in all low-resource settings, particularly at 5% and 10%, where GEMS surpasses it by 5.6% and 6.6%, respectively. These results highlight our model's strong generalization with limited training data.

4.8 Case Study

We conduct a case study to assess GEMS's performance in multi-event extraction. As shown in Table 7, the document presents a scenario with two identical events, which share overlapping arguments "McCarthy". This situation highlights the challenge of extracting event arguments when multiple arguments share the same role within a single event and when arguments are repeated across different events. By leveraging an ontology-driven steering mechanism, GEMS successfully captures the shared characteristics of these overlapping arguments within a semantically enriched token representation space. Without the event-specific context supervision during decoding, GEMS fails to predict "BZW", "Kleinwort Benson", and the overlapping argument "McCarthy" in *Event 1*. Moreover, for multi-word argument terms such as "Department" and "Department of Trade and Industry", GEMS effectively distinguishes between potential spans and accurately defines the boundaries of multi-word argument terms.

Context: As well as previously holding senior positions at Barclays Bank[Entity], BZW[Entity] and Kleinwort Benson[Entity], McCarthy[Person] was formerly a top civil servant at the Department of Trade and Industry[Entity].	
Event1: Personnel:End-Position; Trigger: previously ***Event2:*** Personnel:End-Position; Trigger: formerly	
GEMS	***Event1:*** Barclays Bank[Entity]; BZW[Entity]; Kleinwort Benson[Entity]; McCarthy[Person] ; ***Event2:*** McCarthy[Person] ; Department of Trade and Industry[Entity];
$\text{GEMS}_{\text{w/o steer}}$	***Event1:*** Barclays Bank[Entity]; ~~BZW[Entity]~~; ~~Kleinwort Benson[Entity]~~; ~~McCarthy[Person]~~ ; ***Event2:*** McCarthy[Person] ; Department[Entity] ✗;
$\text{AMPERE}_{\text{Roberta}}$	***Event1:*** Barclays Bank[Entity]; BZW[Entity]; Kleinwort Benson[Entity]; ~~McCarthy[Person]~~ ; ***Event2:*** McCarthy[Person] ; Department of Trade[Entity] ✗; Industry[Entity] ✗;
TagPrime	***Event1:*** ~~Barclays Bank[Entity]~~ ; ~~BZW[Entity]~~; Kleinwort Benson[Person] ✗; ~~McCarthy[Person]~~ ; ***Event2:*** McCarthy[Person] ; Department of Trade and Industry[Entity];
DEEIA	***Event1:*** Barclays Bank[Entity]; BZW[Entity]; ~~Kleinwort Benson[Entity]~~; ~~McCarthy[Person]~~ ; ***Event2:*** McCarthy[Person] ; Department of Trade[Entity] ✗; Industry[Entity] ✗;

Table 7: Case study on ACE05-E with 50% training data. ~~Red crossing line~~ indicates missing predictions. Overlapping argument is Highlighted.

5 Conclusion

In this paper, we present GEMS, a generation-based framework for EAE that leverages Multi-perspective prompts and ontology-guided Steering. By using multiple unanswered prompts to predict event arguments in different orders, GEMS effectively captures the interactions among event elements. The ontology-driven steering mechanism, implemented through a dual-branch cross-attention process, ensures that the generated arguments are contextually relevant and aligned with event-specific knowledge. Extensive experiments on the ACE05-E and ERE-EN datasets show that GEMS outperforms existing state-of-the-art models, particularly in low-resource settings.

Limitation

We acknowledge several limitations of our approach. First, the computational cost of GEMS increases with the number of multi-perspective prompt permutations. Second, GEMS currently lacks an adaptive mechanism for selecting optimal permutations, relying instead on predefined settings. More intelligent strategies could improve efficiency. Finally, GEMS could be further extended to structured information extraction tasks, presenting opportunities for future research.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable discussion and constructive feedback. This work was supported by the National Natural Science Foundation of China (U22B2061), the National Key R&D Program of China (2022YFB4300603) and the Natural Science Foundation of Sichuan, China (2024NSFSC0496).

References

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from bert. In *The 34th AAAI Conference on Artificial Intelligence, AAAI 2020, The 32nd Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The 10th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, volume 34, pages 7456–7463. AAAI Press.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, and 12 others. 2022. Scaling instruction-finetuned language models. abs/2210.11416, CoRR.

Nan Ding, Chunming Hu, Kai Sun, Samuel Mensah, and Richong Zhang. 2022. Explicit role interaction network for event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3475–3485, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020a. Document-level event role filler extraction using multi-granularity contextualized encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8010–8020, Online. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020b. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 671–683, Online. Association for Computational Linguistics.

Zhibin Gou, Qingyan Guo, and Yujiu Yang. 2023. Mvp: Multi-view prompting improves aspect sentiment tuple prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4380–4397. Association for Computational Linguistics.

Chi Han, Jialiang Xu, Manling Li, Yi Fung, Chenkai Sun, Nan Jiang, Tarek Abdelzaher, and Heng Ji. 2024. Word embeddings are steers for language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16410–16430, Bangkok, Thailand. Association for Computational Linguistics.

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

I-Hung Hsu, Kuan-Hao Huang, Shuning Zhang, Wenxin Cheng, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2023a. TAGPRIME: A unified framework for relational structure extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12917–12932, Toronto, Canada. Association for Computational Linguistics.

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

Mengting Hu, Yike Wu, Hang Gao, Yinhao Bai, and Shiwan Zhao. 2022. Improving aspect sentiment quad prediction via template-order data augmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7889–7900, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kuan-Hao Huang, I-Hung Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, and Heng Ji. 2024. TextEE: Benchmark, reevaluation, reflections, and future challenges in event extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12804–12825, Bangkok, Thailand. Association for Computational Linguistics.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020a. How can we know what language models know? 8:423–438, Transactions of the Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020b. How can we know what language models know? 8:423–438, Transactions of the Association for Computational Linguistics.

Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. 2023. Self-regulating prompts: Foundational model adaptation without

forgetting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15190–15200.

Sua Lee, Kyubum Shin, and Jung Ho Park. 2025. Weighted multi-prompt learning with description-free large language model distillation. In *The Thirteenth International Conference on Learning Representations*.

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.

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

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Xiang Li, Zhilei Hu, and 1 others. 2024. Knowcoder: Coding structured knowledge into llms for universal information extraction. In *Proceedings of the 62st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Bangkok, Thailand. Association for Computational Linguistics.

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

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Online, November 16-20, 2020*, pages 1641–1651. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. abs/2107.13586(9):1–35, CoRR.

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

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. In *Findings of the Association for Computational Linguistics: ACL 2024*, Bangkok, Thailand. Association for Computational Linguistics.

Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022. Dynamic prefix-tuning for generative template-based event extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5216–5228, Dublin, Ireland. Association for Computational Linguistics.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Lei Luo and Yajing Xu. 2023. Context-aware prompt for generation-based event argument extraction with diffusion models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 17171725, New York, NY, USA. Association for Computing Machinery.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6759–6774, Dublin, Ireland. Association for Computational Linguistics.

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

Amir Pouran Ben Veyseh, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Graph transformer networks with syntactic and semantic structures for event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3651–3661, Online. Association for Computational Linguistics.

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

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.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton-Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, and 6 others. 2023. Code llama: Open foundation models for code. abs/2308.12950, CoRR.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. Gollie: Annotation guidelines improve zero-shot information-extraction. In *The 20th International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Online, April 19 - 23, 2021*, pages 255–269. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence and 30th Innovative Applications of Artificial Intelligence Conference and 8th AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18, pages 5916–5923. AAAI Press.

Beth M. Sundheim. 1992. Overview of the fourth Message Understanding Evaluation and Conference. In *Proceedings of the 4th Conference on Message Understanding, MUC 1992, McLean, Virginia, USA, June 16-18, 1992*, pages 3–21. Association for Computational Linguistics.

Francesco Tonolini, Nikolaos Aletras, Jordan Massiah, and Gabriella Kazai. 2024. Bayesian prompt ensembles: Model uncertainty estimation for black-box large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12229–12272, Bangkok, Thailand. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. abs/2307.09288, CoRR.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

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

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023. Instructuie: Multi-task instruction tuning for unified information extraction. abs/2304.08085, CoRR.

Jing Xu, Dandan Song, Siu Cheung Hui, Fei Li, and Hao Wang. 2023. Multi-view entity type overdependency reduction for event argument extraction. 265(C), Knowledge-Based Systems.

Yu Yang, Jinyu Guo, Kai Shuang, and Chenrui Mao. 2024. Scented-EAE: Stage-customized entity type

embedding for event argument extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5222–5235, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Jian Zhang, Changlin Yang, Haiping Zhu, Qika Lin, Fangzhi Xu, and Jun Liu. 2024. A semantic mention graph augmented model for document-level event argument extraction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pages 1577–1587, Torino, Italia. ELRA and ICCL.

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

Zixuan Zhang and Heng Ji. 2021. Abstract Meaning Representation guided graph encoding and decoding for joint information extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–49, Online. Association for Computational Linguistics.

Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. 2002. Ensembling neural networks: Many could be better than all. 137(1):239–263, Artificial Intelligence.

Current Term	Candidate Token List
[T]	Input sentence, [SSEP]
[A]	Input sentence, "null", [SSEP]
[R]	All argument roles, [SSEP]
[SSEP]	[T], [A], [R]

Table 8: Candidate token lists of current terms in decoding process.

A Constrained Decoding Scheme

Given an input sentence, we construct multiple prompts and corresponding targets in the same linearized structure to guide the generation format. However, the generated results may not conform to the target schema format, especially in low-resource settings (Lu et al., 2021, 2022; Gou et al., 2023).

To make sure the predicted output complies with the mandatory format, we apply the constrained decoding (CD) algorithm in experiments. Rather than search the whole vocabulary for the next token to decode, which may make the model generate invalid sequences that do not match our expectations, CD adjusts the candidate list dynamically in terms of the current state token by token. If the current token is decoded as "**[**", which means the next token should be selected from a list of terms, i.e., "**[T]**", "**[A]**", "**[R]**" and "**[SSEP]**". Additionally, CD tracks the current term and decodes the next following tokens based on Table 8.

B Baselines and Dataset

B.1 Baselines

We compare GEMS with several state-of-the-art baselines in three categories:

(1) *Multi-label Classification models*:

DyGIE++ (Wadden et al., 2019) performs event argument extraction by scoring spans with contextualized representations, capturing both local and global context for improved task performance.

OneIE (Lin et al., 2020) is a joint event argument extraction framework that incorporates global features to capture cross-task and cross-instance dependencies.

Query and Extract (Wang et al., 2022) uses event types and argument roles as natural language queries, leveraging attention mechanisms to capture semantic correlations and unify event annotations from various ontologies.

Hyperparameter	Values
Batch size	8, 16
Weight decay	0.01
Optimizer	AdamW
Adam ϵ	1×10^{-8}
Max sequence length	250
Permutation number	4
Attention heads a	4

Table 9: Hyperparameters for GEMS across both benchmarks.

AMR-IE (Zhang and Ji, 2021) introduces an AMR-guided framework that captures non-local word connections by aggregating neighborhood information and uses hierarchical decoding based on AMR graph structure.

DEEIA (Liu et al., 2024) extracts event arguments simultaneously using a multi-event prompt mechanism with dependency-guided encoding and event-specific information aggregation to enhance context understanding.

(2) *Sequential Generation models*:

PAIE (Ma et al., 2022) integrates prompt tuning with span selectors for each role, using multi-role prompts and bipartite matching loss for joint optimization to extract event arguments efficiently.

DEGREE (Hsu et al., 2022) is a data-efficient model for low-resource event argument extraction that formulates it as a conditional generation problem, using manually designed prompts to generate event summaries.

AMPERE (Hsu et al., 2023b) combines discrete prompts with resource-enhanced methods, utilizing Abstract Meaning Representation (AMR)-aware prefixes to guide the generation process and improve event argument extraction.

TagPrime (Hsu et al., 2023a) employs structure-generation techniques along with resource-enhanced methods, aiming to unify relational structure extraction tasks through a unified framework.

Scented-EAE (Yang et al., 2024) utilizes discrete prompts to guide event argument extraction, focusing on stage-customized entity type embeddings to enhance non-autoregressive generation.

(3) *LLM-driven models*:

UIE (Lu et al., 2022) is a text-to-structure generation framework that models diverse information extraction tasks, adapts to linearized target event structure using schema-based prompts.

InstructUIE (Wang et al., 2023) is a text-to-structure generation framework built on the instruction-tuning mechanism, leveraging FlanT5-11B[5] (Chung et al., 2022) to model diverse tasks and capture inter-task dependencies.

GoLLIE (Sainz et al., 2024) utilizes Code-LLaMA[6] (Rozière et al., 2023) as its backbone for event argument extraction, refining on human-annotated data to effectively follow annotation guidelines.

LLaMA (Touvron et al., 2023) refers to the direct fine-tuning of LLaMA2-7B[7] on partial training data, following the approach of Li et al. (2024).

KnowCoder (Li et al., 2024) utilizes LLaMA2 for event argument extraction through code generation, incorporating a code-style schema representation and a two-phase learning framework with code pretraining and instruction tuning.

Implementation Details. For the models we re-trained, we keep all hyper-parameters the same with default settings in their original papers. For GEMS, we train model on single NVIDIA-A100 GPU. For each setting, we train models with 5 fixed seeds and 2 learning rates [1e-4, 9e-5]. Then we record the test set performance of the model that performs best on the development set for each random seed. The final reported performance is the average value of results w.r.t five different seeds. We list other important hyperparameters in Table 9. The event-specific ontology description can be accessed at `https://nlp.jhu.edu/schemas/`.

B.2 Datasets

The dataset statistics of `ACE05-E` and `ERE-EN` for main experiments are presented in Table 10. After preprocessing, `ACE05-E` has 33 event types and 22 argument roles, while `ERE-EN` has 38 event types and 21 argument roles in total. we pre-process each dataset by splitting documents into individual sentence. After splitting, `ACE05-E` contains 5,057 events with 6,040 arguments, and `ERE-EN` contains 7,284 events with 10,476 arguments in total. To further investigate the performance of GEMS in low-resource scenarios, we present statistics for subsets of the original dataset, including 5%, 10%, 20%, 30%, and 50% of the data. The statistics on the number of multiple events discussed in Section 4.5 are presented in Table 6.

[5] https://huggingface.co/google/flan-t5-xxl
[6] https://huggingface.co/codellama
[7] https://huggingface.co/meta-llama/Llama-2-7b

Dataset	Split	Docs	Sents	Events	Event Types	Args	Arg Types
ACE05-E	5%	25	649	212	27	228	21
	10%	50	1688	412	28	461	21
	20%	110	3467	823	33	936	22
	30%	160	5429	1368	33	1621	22
	50%	260	8985	2114	33	2426	22
	100%	529	17172	4202	33	4859	22
	Dev	28	923	450	21	605	22
	Test	40	832	403	31	576	20
ERE-EN	5%	20	701	437	31	640	21
	10%	40	1536	618	37	908	21
	20%	80	2848	1231	38	1656	21
	30%	120	4382	1843	38	2632	21
	50%	200	7690	3138	38	4441	21
	100%	396	14736	6208	38	8924	21
	Dev	31	1209	525	34	730	21
	Test	31	1163	551	33	822	21

Table 10: Statistics of datasets used in experiments. "**Split**" denotes the proportion of training data sampled from original datasets.

Event Nums	1	2	3	≥ 4
Train Set (100%)	3966	2787	1225	946
Train Set (20%)	779	483	200	194
Train Set (5%)	229	203	123	85
Dev Set	295	239	91	105
Test Set	334	245	160	83

Table 11: Additional statistics on the number of multiple events in Section 4.5.

C Additional Experimental Analysis

C.1 Analysis on Prompt Permutations and Aggregation Strategies

We provide further analysis on the rank aggregation strategy used during inference. The detailed experimental results are provided in Table 12 and 13. To select the most promising prompt orders, we rank all possible prompts based on their average entropy. Specifically, after constructing a set of input sequences x_{p_i} from the text S as described in Section 3.2, we query the pre-trained language model to obtain the scores for each sequence (Equation 9), based on the word logit for each generated token y_j. Each input sequence x_{p_i} corresponding to permutation p_i is then ranked, and the outputs from the top m-ranked permutations are aggregated to produce the final results.

$$\text{Score}_{rank} = \sum_{j}^{n_y} \log \mathbf{P}\left(y_j | \mathbf{H}_{x_{p_i}}, y_{<j}, f_{CAM}\right), \quad (9)$$

As stated in Section 4.3, no single prompt permutation consistently outperforms the others across varying training data sizes. While the prompt permutation [T][R][A] may outperform other variants, achieving an Arg-I F-1 score of 76.65% under the fully supervised setting, it falls short compared to all other variants when only 30% of the training data is used. This observation further reinforces the effectiveness of our proposed aggregation strategies, with the voting strategy emerging as the most stable and effective approach among the three evaluated strategies.

C.2 Interpretation of Assessment Indicators for Event Steer in Table 4

Here is the supplementary explanation for Table 4. We calculate the standard deviation of the cosine similarity between the output argument word representations. Given n generated argument tokens, where each token's hidden state is denoted as h', the cosine similarity between any two tokens h'_i and h'_j is computed as:

$$c_{i,j} = \frac{h'_i \cdot h'_j}{\|h'_i\| \|h'_j\|}, i \neq j \quad (10)$$

The number of $c_{i,j}$ is $k = n(n-1)/2$. The standard deviation is then computed as:

$$\textit{Std.} = \sqrt{\frac{1}{k} \sum_{i \neq j}^{k} \left(c_{i,j} - \frac{1}{k} \sum_{i' \neq j'}^{k} c_{i',j'} \right)} \quad (11)$$

Additionally, the volume of the bounding boxes surrounding these tokens is computed as:

$$\textit{BBox Vol.} = \prod_{i=1}^{d} (\max(h'[:,i]) - \min(h'[:,i])) \quad (12)$$

where d denotes the dimension of h'.

C.3 Effectiveness on Long-range Dependency in WikiEvent

Event arguments in datasets like ACE05-E and ERE-EN are typically confined to single sentences, limiting the scope of argument extraction to more straightforward cases. However, in real-world scenarios, event arguments often span across multiple sentences, requiring models to capture and link long-range dependencies within the text. To evaluate how well models handle such complex cases, we additionally use the WikiEvent dataset, which

ACE05-E		100%		50%		30%		20%		10%		5%	
		Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
p_i	[T] [A] [R]	76.09	73.98	75.78	72.38	71.79	68.19	71.39	67.56	65.47	60.23	57.95	51.35
	[T] [R] [A]	76.65	74.63	76.75	73.52	71.75	68.47	72.50	69.15	66.23	60.02	56.56	51.54
	[A] [R] [T]	76.11	74.00	75.79	72.38	72.76	69.64	71.62	67.93	66.04	59.77	59.30	53.03
	[R] [A] [T]	76.27	74.05	75.31	72.43	73.09	70.27	71.81	68.07	67.72	61.30	57.59	52.43
Agg	Ours$_{random}$	75.78	73.21	75.98	72.24	71.75	68.47	71.72	67.82	66.79	60.58	57.56	51.31
	Ours$_{rank}$	75.09	72.86	75.89	72.51	72.70	69.66	71.81	67.80	67.04	60.80	56.41	50.68
	Ours$_{vote}$	**76.64**	**74.56**	**76.26**	**72.99**	**73.23**	**69.86**	**71.95**	**67.85**	**67.83**	**61.71**	**59.91**	**53.93**

Table 12: Detailed results for the effect of the each permutation of element order along with different aggregation strategies in ACE05-E dataset (*F1-score,%*).

ERE-EN		100%		50%		30%		20%		10%		5%	
		Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
p_i	[T] [A] [R]	80.20	76.70	79.24	74.48	78.18	73.21	73.88	69.94	75.66	68.97	71.15	66.11
	[T] [R] [A]	79.75	76.25	78.61	73.63	77.99	72.92	73.81	69.43	75.76	68.78	72.46	67.51
	[A] [R] [T]	80.62	77.10	78.80	74.03	77.80	72.81	74.06	70.20	75.61	68.08	71.21	66.41
	[R] [A] [T]	80.47	77.20	78.80	74.03	78.40	72.83	74.13	69.76	75.76	68.53	72.53	67.31
Agg	Ours$_{random}$	79.80	76.67	78.73	73.97	77.97	72.76	73.44	69.18	75.50	68.24	70.98	65.90
	Ours$_{rank}$	79.93	76.67	78.68	73.90	77.55	72.56	74.17	69.65	75.71	68.17	72.09	67.01
	Ours$_{vote}$	**80.22**	**76.76**	**78.77**	**74.10**	**78.47**	**73.40**	**74.41**	**70.35**	**76.22**	**69.08**	**72.14**	**67.17**

Table 13: Detailed results for the effect of the each permutation of element order along with different aggregation strategies in ERE-EN dataset (*F1-score,%*).

WikiEvent	Arg-I	Arg-C
PAIE	69.8	65.2
AMPERE	59.9	53.3
TagPrime	70.3	65.5
DEEIA♣	68.3	64.0
GEMS	**71.6**	**67.6**

Table 14: Performance on passages with long-range dependencies using the WikiEvent dataset (*F1-scores,%*). We reproduce the methods with ♣ by using their released code. The rest are retrieved from Huang et al. (2024). The best results are denoted in **bold**.

consists of passages where event arguments may be distributed across longer text spans.

As shown in Table 14, our model outperforms previous methods by 1.3% in Arg-I F1 and 2.1% in Arg-C F1 as compared to the second-best performing model TagPrime. These improvements demonstrate the ability of GEMS to effectively capture and extract event arguments that extend over long-range dependencies. Additionally, the results on WikiEvent further demonstrate GEMS's potential to generalize across different datasets with varying levels of complexity. While the ACE05-E and ERE-EN datasets are more focused on sentence-level argument extraction, the WikiEvent dataset pushes the boundaries by testing the models ability to handle more intricate, long-range dependencies.

C.4 Argument Identification (Arg-I) Results

Table 15 presents the argument identification results following our main experiments in Table 1, showing the performance of our proposed GEMS model in comparison to other state-of-the-art methods across various data proportions for both the ACE05-E and ERE-EN datasets. For the methods marked with ♣, we re-implemented them using their released code and repeated each experiment setting five times with different random seeds to ensure the reliability of the results. We report the average Arg-I F1-scores for each model under each data proportion. The detailed results show that GEMS also excels in Arg-I F1-score, maintaining strong consistency across different training data proportions.

D Generality of Multi-perspective Prompt Concept

To the best of our knowledge, prompt ensemble or multi-prompt methods in event argument extraction are virtually non-existent, with the exception of GEMS. To investigate the effectiveness of the multi-perspective prompt concept, we explore various single-prompt methods by aggregating predic-

Model	ACE05-E Test Set						ERE-EN Test Set					
	5%	10%	20%	30%	50%	100%	5%	10%	20%	30%	50%	100%
DyGIE++ (2019)	39.2	50.5	57.7	59.9	61.0	63.6	53.3	52.9	55.9	59.1	60.5	63.4
OneIE (2020)	41.3	55.4	64.6	67.8	72.0	73.7	55.5	62.1	67.9	71.9	72.3	75.2
AMR-IE (2021)	43.2	53.3	63.2	67.2	69.5	73.6	47.8	59.1	65.8	71.4	73.9	76.5
Query and Extract (2022)	36.8	33.1	45.6	51.1	56.1	62.4	35.1	46.7	52.1	57.7	64.5	70.4
DEEIA (2024)♣	41.4	55.1	58.4	69.1	72.4	73.8	63.3	66.2	66.5	70.6	73.9	75.6
PAIE (2022)	52.2	62.0	67.8	71.3	72.8	75.0	65.1	71.1	68.7	72.6	74.2	76.6
DEGREE (2022)	47.7	63.0	64.2	70.3	71.4	75.6	66.4	71.2	72.3	74.1	77.4	78.2
AMPERE$_{\text{AMRBart}}$ (2023b)♣	55.4	64.0	70.2	70.8	73.6	73.8	71.3	72.1	72.0	75.2	75.5	77.2
AMPERE$_{\text{AMRRoberta}}$ (2023b)♣	59.7	66.1	66.7	71.4	73.6	75.3	71.5	73.0	72.0	74.0	76.6	77.3
TagPrime (2023a)♣	49.8	63.3	70.3	72.4	74.8	76.1	68.1	70.9	**74.4**	75.9	78.5	79.3
Scented-EAE (2024)♣	58.7	66.2	69.9	72.1	73.2	75.8	67.7	71.9	73.8	75.1	77.5	77.3
GEMS	**59.9**	**67.8**	**72.0**	**73.2**	**76.3**	**76.6**	**72.1**	**76.2**	**74.4**	**78.5**	**78.8**	**80.2**

Table 15: Argument identification results under different data proportion settings for ACE05-E and ERE-EN datasets (*F1-score,%*). The best Arg-I F1-scores are denoted in **bold** and the second highest scores are underlined.

Settings	100%		50%		30%		20%		10%		5%		3%		2%		1%	
	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
Agg$_{random(seed=42)}$	75.56	73.25	71.21	68.42	71.52	67.36	70.63	66.84	63.27	57.43	53.73	48.68	48.22	42.30	34.22	30.08	30.84	27.93
Agg$_{random(seed=3407)}$	73.97	71.30	71.68	69.34	73.79	69.57	69.99	64.64	64.04	58.18	51.47	44.47	49.47	42.40	33.82	28.22	32.89	27.59
Agg$_{vote\geq\frac{3}{4}}$	74.22	72.78	73.26	72.02	73.45	70.72	68.46	67.00	62.00	59.57	47.92	44.93	44.15	42.36	23.68	22.48	18.86	18.83
Agg$_{vote\geq\frac{1}{2}}$	77.11	**74.76**	75.93	**73.14**	75.25	**71.77**	72.45	**69.61**	68.68	**64.32**	58.96	**52.86**	55.89	**51.09**	43.60	**38.80**	40.94	**38.23**
Agg$_{vote\geq\frac{1}{4}}$	72.66	69.44	72.05	69.12	69.92	64.94	68.18	63.46	66.57	59.47	59.59	50.33	55.14	46.18	46.18	38.07	48.52	39.10

Table 16: Performance on models aggregation under different data proportion settings for ACE05-E dataset (*F1-score,%*). The best Arg-C F1-scores are denoted in **bold**.

Settings	100%		50%		30%		20%		10%		5%		3%		2%		1%	
	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
Agg$_{random(seed=42)}$	79.20	75.39	75.45	71.49	75.82	71.62	72.58	69.12	70.47	64.93	67.74	61.72	62.43	57.43	53.69	49.85	36.62	31.96
Agg$_{random(seed=3407)}$	76.81	73.35	74.17	69.88	75.31	71.85	71.95	67.48	72.3	67.32	66.84	61.91	62.64	57.47	53.29	48.59	36.36	31.71
Agg$_{vote\geq\frac{3}{4}}$	78.29	76.15	75.07	71.53	74.91	72.16	71.56	69.36	71.24	67.00	66.67	64.44	60.44	57.34	47.97	46.15	25.79	23.94
Agg$_{vote\geq\frac{1}{2}}$	79.90	**76.92**	77.88	**73.32**	77.35	**73.50**	75.68	**71.50**	75.31	**69.69**	72.90	**66.50**	67.06	**61.48**	60.56	**56.16**	42.21	35.67
Agg$_{vote\geq\frac{1}{4}}$	78.37	74.76	76.92	72.17	75.03	70.37	73.55	68.12	72.41	65.31	69.57	62.94	64.20	57.06	60.51	53.94	46.94	**37.91**

Table 17: Performance on models aggregation under different data proportion settings for ERE-EN dataset (*F1-score,%*). The best Arg-C F1-scores are denoted in **bold**.

tions from AMPERE$_{\text{AMRBart}}$, AMPERE$_{\text{AMRRoberta}}$, TagPrime, and Scented-EAE using different aggregation strategies on the ACE05-E and ERE-EN datasets. Each model was trained three times using three different random seeds, and the average argument prediction results were taken as the model's output. The experimental results are presented in Tables 16 and 17. Although introducing multiple models increases deployment and computational costs significantly, the performance achieves a new state-of-the-art level. This demonstrates the versatility and synergistic potential of the multi-perspective prompt approach.