Scented-EAE: Stage-Customized Entity Type Embedding for Event Argument Extraction

Yu Yang*, **Jinyu Guo*** , **Kai Shuang** † and **Chenrui Mao**
State Key Laboratory of Networking and Switch Technology,
Beijing University of Posts and Telecommunications
{yy-potato, guojinyu, shuangk, maochenrui1218}@bupt.edu.cn

Abstract

Existing methods for incorporating entities into EAE rely on prompts or NER. They typically fail to explicitly explore the role of entity types, which results in shallow argument comprehension and often encounter three issues: (1) weak semantic associations due to missing role-entity correspondence cues; (2) compromised semantic integrity from abandoning context after recognizing entities regardless of their types; (3) one-sided semantic understanding relying solely on argument role semantics. To tackle these issues, we propose Scented-EAE, an EAE model with stage-customized entity type embedding to explicitly underscore and explore the role of entity types, thus intervening in argument selection. Specifically, at the input stage, we strengthen semantic associations by prompting role-entity correspondence after extending a non-autoregressive decoder as part of the encoder. At the intermediate stage, we preserve semantic integrity by optimizing our proposed BIO-aware NER and EAE via a novel IPE joint learning. At the output stage, we expand semantic understanding dimensions by determining arguments using span selectors from argument roles and entity types. Experiments show that our model achieves state-of-the-art performance on mainstream benchmarks. In addition, it also exhibits robustness in low-resource settings with the help of prompts and entity types.[1]

1 Introduction

Event Extraction (EE) is a vital task in natural language processing (NLP), which usually consists of two main subtasks: Event Detection (ED) and Event Argument Extraction (EAE) (Peng et al., 2023). Given that significant progress has been made in ED through prior research (Liu et al., 2022; Guan et al., 2023; Liu et al., 2023b), this paper aims to address the enduring challenges in EAE.

*These authors contributed equally to this work.
†Correponding author.
[1]https://github.com/yy-degit/Scented-EAE

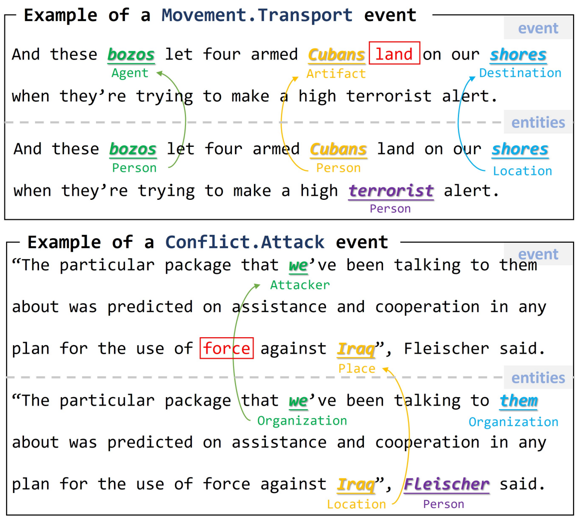

Figure 1: Examples of events where red boxes denote triggers and arrows denote role-entity correspondence.

EAE focuses on extracting arguments for each argument role in the event from the text. As can be observed in Figure 1, in the *Movement.Transport* event triggered by *land*, the *Agent* is "bozos," the *Artifact* is "Cubans," and the *Destination* is "shores." Recent studies (Wang et al., 2022; Hsu et al., 2022; Liu et al., 2023a; Zhang et al., 2023) indicate that incorporating entity knowledge into EAE can improve its performance, which is primarily achieved through prompts or Named Entity Recognition (NER) techniques. However, it is crucial to highlight that both prompts and NER fail to explicitly underscore and explore the role of entity types, causing a lack of in-depth perception of entity type semantics. This deficiency often leads to a shallow and inadequately enriched comprehension of arguments, ultimately reducing the accuracy and generalization of argument extraction.

Hence, we consider explicitly introducing entity types into each stage of EAE to direct the model's attention towards content that aligns with entity type semantics related to argument roles. This in-

Findings of the Association for Computational Linguistics: ACL 2024, pages 5222–5235
August 11-16, 2024

tervention helps filter out more plausible results from the outset, thus mitigating potential errors and simplifying the selection process. To maximize the effectiveness of this introduction, we focus on the existing issues at each stage due to the insufficient incorporation of entity types. Specifically, (1) At the input stage, prompts fail to include genuine entity type labels and role-entity correspondence in argument slots, often resulting in **weak semantic associations** between labels; (2) At the intermediate stage, NER tends to discontinue attention to complete context after determining candidate argument spans, usually leading to **compromised contextual semantic integrity**; (3) At the output stage, general argument selection solely relies on understanding argument role semantics while neglecting the comprehension in entity type dimension, bringing about **one-sided semantic understanding** of decisions. The above issues eventually hinder EAE models from reaching better performance.

In this paper, we propose **Scented-EAE**, an EAE model with **S**tage-**C**ustomized **En**tity **T**ype **E**mbe**d**ding. Based on the characteristics of each stage in EAE, our model customizes the integration of entity types to solve the aforementioned three problems, respectively. Concretely, at the input stage, we aim to strengthen semantic associations through event-specific prompts with argument role labels, entity type labels, and role-entity correspondence. The intuition behind this is that prompting these labels and correspondence in argument slots can enrich argument semantics and enhance the interaction between roles and entities. For instance, when matching arguments for the "Destination" role, entities of the "Location" type are more suitable than those of the "Person" type. Additionally, to further capture the relationship between context and prompts when encoding the inputs, we extend a non-autoregressive decoder as part of the encoder to reinforce feature awareness.

At the intermediate stage, we optimize a BIO-aware NER and EAE via "**I**mplicit **P**lus **E**xplicit (**IPE**)" joint learning to preserve semantic integrity. "Implicit" refers to regular multi-task knowledge sharing, while "Explicit" involves co-learning specific BIO tag parameters for both BIO-aware NER and EAE. Through IPE joint learning, our model enables each token to sense its BIO tag semantics, thus enlarging semantic gaps and dilimiting boundaries between arguments of different entity types.

At the output stage, we extend semantic understanding dimensions by using argument span selectors in two dimensions: argument roles and entity types. Sparked by the empirical evidence that multiple model voting outperforms a single one (Gou et al., 2023), we make a joint decision on argument answers from both comprehension dimensions to enhance their accuracy.

Experiments show that Scented-EAE achieves state-of-the-art performance on three mainstream benchmarks. Our base model reaches an improvement of +2.9%, +3%, and +2% F1 score over other base-scale models, while our large model obtains +1.8%, +2.2%, and +0.9% F1 gains over other large-scale ones. It also exhibits robustness in low-resource settings due to prompts and entity types.

In summary, our contributions are as follows:

- We propose Scented-EAE, an EAE model with stage-customized entity type embedding to explicitly underscore and explore the role of entity types. This integration enriches the comprehension of arguments, thus optimizing the argument selection process and mitigating potential errors.
- We present a novel IPE joint learning to co-train a BIO-aware NER and EAE in an "Implicit Plus Explicit" manner, facilitating argument boundary division and contextual integrity preservation.
- Experiments show that our model achieves state-of-the-art performance on three benchmarks and exhibits robustness in low-resource settings with the help of prompts and entity types.

2 Methodology

As illustrated in Figure 2, our model is transformer-based (Vaswani et al., 2017) and consists of three core modules: *context-template encoding*, *IPE joint learning*, and *span selector decoding*. Details of each module will be provided sequentially.

2.1 Context-Template Encoding

To begin with, we define our dataset on EAE tasks as $\mathcal{D} = \{(\mathcal{C}_i, \mathcal{E}_i, et_i, tr_i, \mathcal{A}_i) | 1 \leq i \leq |\mathcal{D}|\}$, where $\mathcal{C}_i = \{c_{ij} | 1 \leq j \leq |\mathcal{C}_i|\}$ denotes the context, $\mathcal{E}_i = \{(e_{ij}, span_{ij}) | 1 \leq j \leq |\mathcal{E}_i|\}$ denotes the entities (e_{ij} is the entity type and $span_{ij}$ is the entity offset), et_i denotes the event type, tr_i denotes the trigger word, and $\mathcal{A}_i = \{(r_{ij}, span_{ij}) | 1 \leq j \leq |\mathcal{A}_i|\}$ denotes the arguments (r_{ij} is the argument role and $span_{ij}$ is the argument offset).

2.1.1 Template Design

For each instance $(\mathcal{C}, \mathcal{E}, et, tr, \mathcal{A}) \in \mathcal{D}$, we design an appropriate template $\mathcal{T}$ to prompt the semantic relationship between components of the event. As

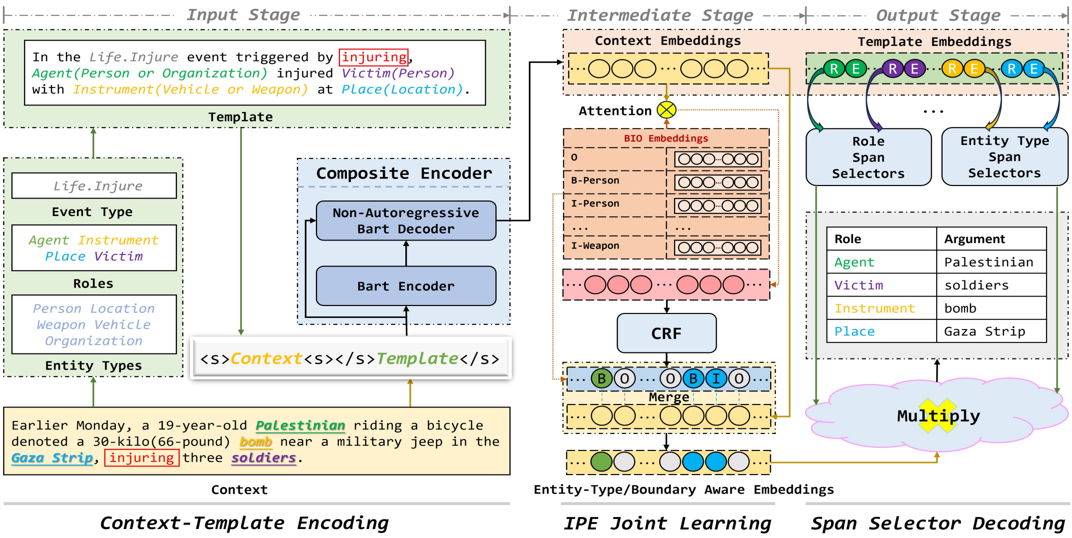

Figure 2: The architecture of Scented-EAE. At the input stage, our model encodes the concatenation of context and template using a composite encoder, generating two parts of embeddings. Then context embeddings are used for IPE joint learning, while template embeddings play a role in acquiring span selectors and decoding arguments.

depicted in Figure 2, our template usually contains four main parts: event type, trigger word, argument role labels, and corresponding entity type labels. To represent the interaction between argument slots in the template, we follow the approach of Ma et al. (2022), utilizing event-specific natural language to connect all argument role labels. For semantic enrichment of slots, we append explanations after each argument role to indicate its corresponding entity types, which are connected by "or" and enclosed in parentheses. This enables the enhanced perception of role-entity correspondence without disrupting dependencies among distinct slots and approaches to natural language expression. Then a concise sentence structure is employed to link the event type and trigger word. The final template for the input in Figure 2 is as follows:

In the Life.Injure event triggered by injuring, Agent (Person or Organization) injured Victim (Person) with Instrument (Vehicle or Weapon) at Place (Location).

Each underlined string works as a part of the event. More templates can be found in Appendix A.1.

It is noteworthy that in our template, each argument role appears only once, and a single argument slot can pay attention to multiple entity types. Unlike Ma et al. (2022), which matches only one argument per slot, we can extract multiple arguments simultaneously using just one slot with the help of threshold tuning. Moreover, our one-to-many ties between argument roles and entity types facilitate the expansion of argument semantics, contributing to a relatively precise understanding. Details about role-entity mapping are shown in Appendix A.2. Additionally, we provide several examples of diverse template designs in Table 1, and associated analysis can be found in Section 4.1.

2.1.2 Interaction-Enhanced Encoding

At the input stage, given the context $\mathcal{C}$ and template $\mathcal{T}$, we initially concatenate them into a sequence following the specific format:

$$\mathcal{X} = [< \mathrm{s} >; \mathcal{C}; < \mathrm{s} >; < /\mathrm{s} >; \mathcal{T}; < /\mathrm{s} >] \quad (1)$$

where $< \mathrm{s} >$ and $< /\mathrm{s} >$ are start and end symbols of BART (Lewis et al., 2020). Using them in pairs helps to seperate the two input parts. Then we feed $\mathcal{X}$ into a composite encoder comprising a classic encoder and a non-autoregressive decoder for one-step encoding, obtaining context embeddings C and template embeddings T as shown below:

$$C, T = \mathrm{CompositeEncoder}(\mathcal{X}) \quad (2)$$

In the composite encoder, $\mathcal{X}$ is concurrently input into both the BART Encoder and the non-autoregressive BART Decoder. This practice aims to reinforce the interaction between context and template through various attention mechanisms, including self-attention, masked-attention, and cross-attention. In contrast to the conventional approach of supplying templates to an autoregressive decoder, our method avoids error propagation during inference and establishes bidirectional dependencies between context and template. A comparative

Type	Example
Scented-EAE	In the Life.Injure event triggered by injuring, Agent (Person or Organization) injured Victim (Person) with Instrument (Vehicle or Weapon) at Place (Location).
ET1	The event type is Life.Injure, the trigger word is injuring, and the role template is ...
ET2	Event: Life.Injure; Trigger: injuring; Role Template: ...
AE1	Person or Organization (Agent) injured Person (Victim) with Vehicle or Weapon (Instrument) at Location (Place).
AE2	Agent, Person or Organization injured Victim, Person with Instrument, Vehicle or Weapon at Place, Location.

Table 1: Variants of templates. Scented-EAE: default template. ETx: modified templates for concatenating event type and trigger word. AEx: modified templates for concatenating argument roles and entity types.

analysis of other encoding strategies will be presented in Section 4.1.

2.2 IPE Joint Learning

Directly querying the context for arguments often leads to inaccuracies in identifying span boundaries, while determining argument boundaries via pipelined NER introduces error accumulation and compromises contextual integrity. To solve these problems, we propose IPE joint leaning at the intermediate stage.

2.2.1 BIO-Aware NER

Existing joint learning typically achieves implicit knowledge sharing through multi-task optimization of losses. Building upon this, we augment the joint learning of NER and EAE by explicitly training specific parameters related to entity types. At first, we refine NER with a unique sequence labeling method to learn entity type semantics after obtaining the context embeddings C. By adopting the cross attention mechanism on a BIO tag embedding matrix M, our BIO-aware NER can realize closer semantic alignment between each contextual token and its golden BIO tag. The resulting context embeddings C_{abio}, which incorporate awareness of all BIO tags, are as follows:

$$C_{abio} = \text{CrossAttention}(C, M, M) \tag{3}$$

where $M \in R^{|\{BIO\ tags\}| \times d_m}$ and d_m is the hidden size of the model.

In order to retain the original information in the context, we merge C and C_{abio} into C_{sl}, which acts as the input of sequence labeling:

$$C_{sl} = \text{GELU}([C; C_{abio}; C \circ C_{abio}]W_1)W_2 \tag{4}$$

where $\circ$ represents the element-wise multiplication, $W_1 \in R^{3d_m \times d_m}$ and $W_2 \in R^{d_m \times |\{BIO\ tags\}|}$ are two learnable matrices, and GELU (Hendrycks and Gimpel, 2016) is an activation function. We adopt a Conditional Random Field (CRF) (Lafferty et al., 2001) layer on C_{sl} to search for the highest-scoring label sequence. During training, we optimize BIO-aware NER with the standard CRF loss function (Chiu and Nichols, 2016), which is denoted as L_{ner}.

2.2.2 Argument Boundary Marking

In EAE with pipelined NER, once candidate argument spans are determined, the remaining context will be treated as irrelevant content and discarded. It causes a loss of complete contextual information. To ensure contextual integrity while partitioning argument boundaries, we explicitly integrating the BIO tag matrix from BIO-aware NER, which represents entity type semantics, into the context of EAE. This integration helps to widen semantic gaps between arguments of different entity types, indirectly marking and distinguishing argument boundaries.

For each token, we retrieve its corresponding BIO tag embeddings from matrix M and aggregate them into a sequence called C_{pbio}. By sharing M between C_{abio} in NER and C_{pbio} in EAE, our model can discern the semantic distinctions among arguments with different BIO tags during sequence labeling and amplify their discrepancies in argument extraction. Upon embedding BIO tags into the context, we obtain the context embeddings C_{entity} that are aware of entity types:

$$C_{entity} = \text{GELU}([C; C_{pbio}]W_3)W_4 \tag{5}$$

where $W_3 \in R^{2d_m \times d_m}$ and $W_4 \in R^{d_m \times d_m}$ are learnable matrices.

2.3 Span Selector Decoding

At the output stage, we make a joint decision on argument selection from the semantic understanding dimensions of argument roles and entity types for more precise decoding.

2.3.1 Role Span Selector

Scented-EAE decodes argument spans through queries from templates like Ma et al. (2022). Given the template embeddings T, we construct span selectors S_r^{start} and S_r^{end} for each role r to identify the start and end positions of arguments. Acquiring the role embeddings ϕ_r in the template, we have:

$$S_r^{start} = \phi_r \circ W_{start1} \tag{6}$$

$$S_r^{end} = \phi_r \circ W_{end1} \tag{7}$$

where $W_{start1} \in R^{1\times d_m}$ and $W_{end1} \in R^{1\times d_m}$ are learnable matrices shared among all roles. Given C_{entity}, we can calculate the probability distribution for selecting each token as the start/end point of argument spans corresponding to role r:

$$\text{logit}_r^{start} = \sigma(S_r^{start} \cdot C_{entity}^{\mathrm{T}}) \tag{8}$$

$$\text{logit}_r^{end} = \sigma(S_r^{end} \cdot C_{entity}^{\mathrm{T}}) \tag{9}$$

where T represents the transpose operation and σ denotes the sigmoid function.

2.3.2 Entity Type Span Selector

Similarly, we design entity type span selectors $S_{e(r)}^{start}$ and $S_{e(r)}^{end}$ using $\phi_{e(r)}$, which are the average embeddings of all tokens within the entity types $e(r)$ corresponding to role r in the template:

$$S_{e(r)}^{start} = \phi_{e(r)} \circ W_{start2} \tag{10}$$

$$S_{e(r)}^{end} = \phi_{e(r)} \circ W_{end2} \tag{11}$$

where $W_{start2} \in R^{1\times d_m}$ and $W_{end2} \in R^{1\times d_m}$ are learnable matrices. Entity type span selectors are also used to calculate the probability distribution for selecting spans:

$$\text{logit}_{e(r)}^{start} = \sigma(S_{e(r)}^{start} \cdot C_{entity}^{\mathrm{T}}) \tag{12}$$

$$\text{logit}_{e(r)}^{end} = \sigma(S_{e(r)}^{end} \cdot C_{entity}^{\mathrm{T}}) \tag{13}$$

The final distribution of argument span selection is computed by combining the results of role span selectors and entity type span selectors:

$$\text{logit}^{start} = \text{logit}_r^{start} \circ \text{logit}_{e(r)}^{start} \tag{14}$$

$$\text{logit}^{end} = \text{logit}_r^{end} \circ \text{logit}_{e(r)}^{end} \tag{15}$$

Then we define the loss function in EAE as:

$$L_{eae} = \sum_{i=1}^{|\mathcal{D}|}\sum_{j=1}^{|\mathcal{R}_i|}\sum_{k=1}^{|\mathcal{C}_i|}\Big(L_{bce}\left(\text{logit}_j^{start}(k), y_k^{start}\right) + L_{bce}\left(\text{logit}_j^{end}(k), y_k^{end}\right)\Big) \tag{16}$$

$$L_{bce}(x,y) = -(y\text{log}x + (1-y)\text{log}(1-x)) \tag{17}$$

where $\mathcal{D}$ is the dataset, $\mathcal{R}$ is the role set, $\mathcal{C}$ is the context, logit^{start} and logit^{end} are final distributions for span selection, y^{start} and y^{end} are golden labels of argument start and end positions, and L_{bce} is the standard BCEloss, where x represents the score and y represents the golden label. Above all, the final loss function of Scented-EAE is:

$$L_{Scented\text{-}EAE} = L_{ner} + L_{eae} \tag{18}$$

During inference, we predict legitimate and nearest argument spans that exceed the start/end thresholds for all roles in parallel.

3 Experiment

In this section, we conduct experiments to evaluate the performance of Scented-EAE.

3.1 Experimental Setup

Datasets We conduct experiments on **ACE05-E**, **ACE05-E$^+$** (Doddington et al., 2004), and **ERE**[2] (Song et al., 2015), which offer event and entity annotations suitable for evaluating sentence-level EAE tasks. Following the approaches of Wadden et al. (2019) and Lin et al. (2020), we preprocess ERE and transform ACE2005 into two variants: ACE05-E and ACE05-E$^+$. ACE2005 contains 33 event types and 22 argument roles, while ERE includes 38 event types and 21 argument roles. More dataset details are provided in Appendix A.3.

Baselines We compare Scented-EAE with the following competitive models: (1) **EEQA** (Du and Cardie, 2020) sets dynamic thresholds to extract arguments based on QA. (2) **OneIE** (Lin et al., 2020) utilizes global features to perform joint information extraction. (3) **PAIE** (Ma et al., 2022) chooses the best argument spans via prompt tuning. (4) **DEGREE** (Hsu et al., 2022) leverages weak supervision signals for strong performance in low-resource settings. (5) **TabEAE (Multi-Single)** (He et al., 2023) realizes single-event argument inference through multi-event training. (6) **AMPERE (RoBERTa)** (Hsu et al., 2023) introduces Abstract Meaning Representation (AMR) into prefix encoded by RoBERTa (Liu et al., 2019) in generative models. We create fitting templates for ERE in EEQA, PAIE, and TabEAE. Official codes of all baselines except for OneIE are trained with default parameters. OneIE is modified to be a version suitable for EAE tasks.

[2] https://catalog.ldc.upenn.edu/LDC2023T04

Model	PLM	ACE05-E		ACE05-E$^+$		ERE	
		Arg-I	Arg-C	Arg-I	Arg-C	Arg-I	Arg-C
EEQA	BERT-b	68.2	65.4	66.5	64.5	72.8	66.4
(Du and Cardie, 2020)	BERT-l	70.5	68.9	67.1	65.2	72.4	67.6
OneIE	BERT-b	65.9	59.2	63.9	60.0	66.8	58.3
(Lin et al., 2020)	BERT-l	73.2	69.3	73.3	70.6	68.2	64.2
PAIE	BART-b	73.0	70.6	74.1	69.2	76.2	69.6
(Ma et al., 2022)	BART-l	75.7	73.3	75.9	72.5	78.4	71.2
DEGREE	BART-b	73.5	69.0	72.0	67.9	75.6	70.0
(Hsu et al., 2022)	BART-l	76.0	73.5	75.2	73.0	76.2	73.2
TabEAE (Multi-Single)	RoBERTa-b	72.9	69.6	71.2	68.1	76.5	70.8
(He et al., 2023)	RoBERTa-l	76.3	73.5	75.5	72.6	78.8	73.2
AMPERE (RoBERTa)	BART-b	73.7	70.9	74.6	70.5	75.2	70.1
(Hsu et al., 2023)	BART-l	76.2	73.8	76.0	73.0	76.4	72.1
Scented-EAE	BART-b	76.3	73.8	76.9	73.5	76.0	72.8
	BART-l	**77.1**	**75.6**	**77.4**	**75.2**	**79.7**	**74.1**

Table 2: F1 scores of Scented-EAE and all baselines on EAE tasks. We **bold** the best result and underline the second best. b in column PLM denotes base models and l denotes large models.

Evaluation Metrics Dominant evaluation metrics are adopted: (1) **Arg-I**: An argument is correctly identified if the predicted span matches with a golden span. (2) **Arg-C**: An argument is correctly classified if it is correctly identified and the predicted role matches with a golden role.

Implementation Details We initialize the composite encoder using BART-base and BART-large, respectively. The learning rate of parameters ranges from 1e-5 to 1e-4. The selection thresholds for argument start/end positions vary from 0 to 1. We optimize our model on NVIDIA RTX 3090 GPU by AdamW (Loshchilov and Hutter, 2017). More hyperparameter details are shown in Appendix A.4.

3.2 Main Results

From Table 2, we can observe that Scented-EAE outperforms all the other baselines across three benchmarks. Concretely, our base model reaches an improvement of +2.9%, +3%, and +2% in Arg-C F1 score over other base-scale models, while our large model obtains +1.8%, +2.2%, and +0.9% Arg-C F1 gains over other large-scale ones. This indicates that our model consistently demonstrates superior performance across all fair comparison scenarios with other methods.

We also find that the performance of our base model even surpasses that of most large baselines. This phenomenon suggests that our designs empower base scaled-parameter models to achieve comparable efficacy to larger ones, thereby conserving computational resources. We attribute this capability to the comprehensive exploitation of limited entity type knowledge by Scented-EAE. Furthermore, it should be pointed out that baselines relying on prompts (PAIE and TabEAE) or external knowledge (DEGREE and AMPERE) often yield suboptimal results. However, in contrast to their superficial incorporation of various prompts or external information, thoroughly exploring the effectiveness of entity types can lead to superior performance.

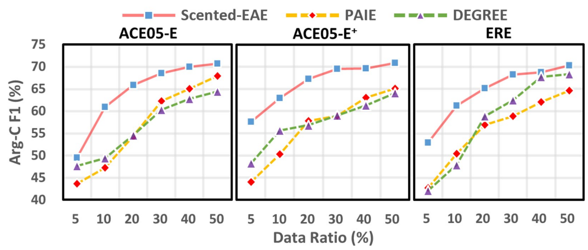

Figure 3: Arg-C F1 in low-resource settings.

3.3 Low-resource Settings

To assess the proficiency of Scented-EAE in scenarios with a scarcity of data annotations, we conduct low-resource experiments by randomly selecting 5%-50% samples for training. Then we compare the results of Scented-EAE, PAIE, and DEGREE on the complete test set. Notably, from this point on, all subsequent experiments are conducted using the **BART-base** pre-trained language model.

Template	ACE05-E	ACE05-E$^+$	ERE
Scented-EAE	**73.8**	**73.5**	**72.8**
ET1	72.4	72.8	70.9
ET2	73.5	72.3	70.4
AE1	73.5	72.9	71.2
AE2	73.4	72.8	71.1
PREFIX	71.3	70.8	71.0

Table 3: Arg-C F1 with different template designs.

Method	ACE05-E	ACE05-E$^+$	ERE
Scented-EAE	**73.8**	**73.5**	**72.8**
BART-S	72.2	71.6	71.9
BERT-C	70.6	69.7	70.2
BERT-S	63.1	62.4	63.6

Table 4: Arg-C F1 with different encoding mechanisms.

Figure 3 illustrates that Scented-EAE outperforms the other two models comprehensively, revealing its capability to learn well even with limited training data under conditions where both prompts and entity types act as weak supervision signals.

4 Ablation and Analysis

In this section, we analyze the effect of different proposed techniques in Scented-EAE.

4.1 Model Variable Substitution

Template Design We consider template concatenation designs from three aspects: (1) Event type and trigger word: **ET1** and **ET2**. (2) Argument roles and entity types: **AE1** and **AE2**. (3) Template and context: **PREFIX** and **Scented-EAE**. Among them, Scented-EAE, (1), and (2) are exemplified in Table 1. PREFIX and Scented-EAE in (3) represent the concatenation methods of placing the template before and after context. Table 3 demonstrates that PREFIX degrades the model performance. We argue that when placing the template before context, the model start symbol is farther away from context, leading to insufficient perception of overall contextual information and decreased effect. Besides, Scented-EAE is the best due to its close resemblance to natural language expressions. This is also verified by the fact that the design for exchanging positions of argument roles and entity types yields suboptimal results.

Encoding Mechanism We compare four encoding mechanisms: (1) **Scented-EAE**: encoding the concatenated context and template in our composite encoder. (2) **BART-S**: encoding context in encoder and decoding template in decoder. (3) **BERT-C**: encoding the concatenated context and template in the encoder. (4) **BERT-S**: encoding context and template sequentially in the encoder. (1) and (2) are initialized by BART, yet (3) and (4) are initialized by BERT (Devlin et al., 2019). We discover from Table 4 that concatenated encoding is superior to separate encoding, and encoder-decoder architectures are better than encoder-only ones. It implies the benefit of bidirectional interaction between context and template and feature awareness through various attention mechanisms. Our approach of extending a non-autoregressive decoder to be part of the encoder works well with all these advantages.

Method	ACE05-E	ACE05-E$^+$	ERE
Scented-EAE	**73.8**	**73.5**	**72.8**
Implicit Joint Learning	72.8	71.9	71.5
Piepelind NER	71.6	70.8	69.5

Table 5: Arg-C F1 with different NER methods.

Model	ACE05-E	ACE05-E$^+$	ERE
Scented-EAE	**73.8**	**73.5**	**72.8**
-w/o Entity Type Prompt	71.5	71.9	70.9
-w/o IPE Joint Learning	71.8	70.5	70.1
-w/o Entity Type Selector	72.5	71.5	70.7

Table 6: Ablation results.

NER Method To test the validity of our IPE joint learning, we set up the following three comparisons: (1) **Scented-EAE**: "Implicit Plus Explicit" joint learning. (2) **Implicit Joint Learning**: classical joint learning (Lin et al., 2020). (3) **Pipelined NER**: NER for direct role matching after determining candidate argument sets. As can be seen from Table 5, Pipelined NER that utilizes only entity mentions regardless of contextual integrity suffers from error accumulation and information loss, causing poor results. While classical Implicit Joint Learning does not destroy context integrity, the lack of our explicit introduction of entity types to mark argument boundaries also makes it deteriorates.

4.2 Core Component Ablation

We further examine the necessity of our critical designs through ablation experiments, including: (1) **w/o Entity Type prompt**: removing entity type prompts from the template. (2) **w/o IPE Joint Learning**: removing IPE Joint Learning with BIO-aware NER. (3) **w/o Entity Type Selector**: removing entity type span selectors. The results of the ablation experiments are summarized in Table 6. We can find that each module of Scented-EAE that customizes the integration of entity type knowledge into EAE contributes to notable performance improvements. These improvements are particularly apparent in ACE05-E$^+$ and ERE datasets, which encompass complex arguments with a higher number of pronouns and lengthier texts. It can be generalized that our model can also parse complex event structures well by sensing entity type semantics.

Although different datasets exhibit varying sensitivities to distinct designs, overall, IPE joint learning plays the most important role among all designs. It enables the perception of entity type semantics without contextual integrity loss during joint learning and helps enlarge semantic gaps between arguments with different entity types to distinguish argument boundaries. Additionally, by employing argument role and entity type dual span selectors to determine argument spans, we can obtain more comprehensive and precise interpretations of argument expressions. Moreover, template prompts for entity types and role-entity correspondence also demonstrate great behaviors. Such prompts can be aware of entity type semantics and capture role-entity associations to enhance understanding and dependencies of arguments.

4.3 Case Study

Figure 4 illustrates two test cases of PAIE and Scented-EAE. In Example 1, PAIE incorrectly predicts "U.N." as the argument for the "Entity" role in the "Contact.Meet" event, while Scented-EAE accurately extracts the complete argument "U.N. Security Council." This showcases the efficacy of our entity type incorporation approach in distinguishing argument boundaries. In Example 2, PAIE erroneously identifies "tribunal" as the argument for the "Destination" role in the "Justice.Extradite" event, despite it being an unreasonable "Organization" entity type. In contrast, Scented-EAE circumvents this issue by leveraging the prompt of correspondence between the "Destination" role and the "Location" entity type.

4.4 Application to LLMs

To evaluate the effectiveness of our design on large language models (LLMs), we conduct similar experiments using T5-3B [3] and LLAMA3-8B [4]. The results in Appendix A.5 indicate that due to the current limitations in the amount of training data, we have not fully harnessed the capabilities of LLMs.

[3]https://huggingface.co/google/t5-v1_1-xl

[4]https://huggingface.co/meta-llama/Meta-Llama-3-8B

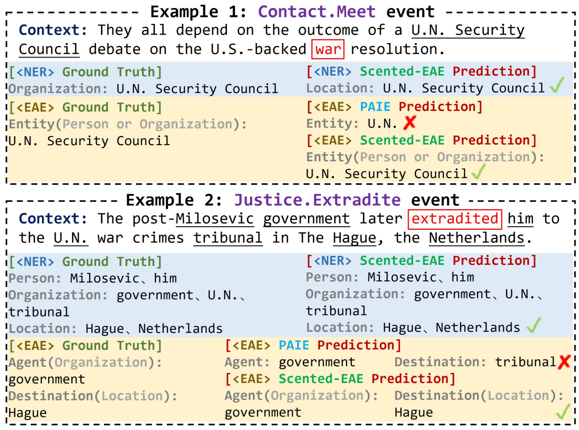

Figure 4: Two test cases from ACE05-E$^+$ where grey words are entity types or argument roles (corresponding entity types are in parentheses).

Consequently, LLM results decline somewhat compared to traditional pre-trained language models. However, we believe that with more extensive data, there is substantial potential for improvement. In addition, T5-3B with an encoder-decoder architecture outperforms LLAMA3-8B with a decoder-only architecture. This suggests that the various attention mechanisms in encoder-decoder architecture help extract more precise embedding features.

5 Related Works

Template-Guided EAE: Whether for document-level EAE tasks (Li et al., 2021) or sentence-level ones (Ma et al., 2023), predicting arguments based on templates is popular. Templates are designed in various ways such as directly linking argument slots (Wang et al., 2022) or connecting argument role labels with natural language (Ma et al., 2022). Argument slots are often denoted by placeholders (Li et al., 2021), argument roles (Ma et al., 2022), or related entity descriptions (Hsu et al., 2022). Besides, some studies (Lu et al., 2021, 2022) have proposed templates in forms of linearization representations, which also achieve promising results. Recent research has introduced Abstract Meaning Representation (AMR) (Yang et al., 2023; Xu et al., 2023), prefix tuning (Liu et al., 2022; Hsu et al., 2023), and event co-occurrence knowledge (He et al., 2023) into template-guided EAE, aiming to enrich semantics and save parameters of models. However, none of these templates provide entity types and role-entity correspondence like Scented-EAE, thus lacking partial semantic dependencies.

NER-Introduced EAE: Document-level EAE (Zheng et al., 2019; Xu et al., 2021; Zhu et al.,

2022) often performs NER first, followed by matching entities with argument roles. For sentence-level EAE, joint information extraction models (Wadden et al., 2019; Liu et al., 2022; Lin et al., 2020; Nguyen et al., 2021) have achieved promising outcomes through knowledge sharing. Recent studies (Zhang et al., 2023; Wang et al., 2023) have made new attempts to introduce NER into EAE. Zhang et al. (2023) proposes cross-dataset transfer learning for pseudo-NER, generating event-related entities before arguments. However, it pays attention solely to argument entities without learning non-argument entity semantics. Wang et al. (2023) employs in-context learning on Large Language Models (LLMs) to infer arguments and entities in forms of Python codes. Nevertheless, generating code-formed outputs through LLMs is time-consuming and costly. Compared to these methods, in our model, all entities in datasets can be noticeable, and the training process costs low resources.

6 Conclusion

In this paper, we propose Scented-EAE, an EAE model with stage-customized entity type embedding to intervene in argument selection and mitigate potential errors. This model achieves state-of-the-art performance on three benchmarks and also shows robustness in low-resource settings. In the future, from a strategic perspective, we anticipate the potential for extending the concept of tailoring integrated, domain-specific knowledge functionalities across various domains. At a granular level, our focus lies in integrating additional attributes of entities into EAE, including but not limited to entity relationships.

Limitations

This paper aims to explore the efficacy of customizing the integration of entity types at each stage of EAE. While our methods have demonstrated strong performance in EAE, they still encounter certain limitations. Specifically, we approach the EAE task through a classification model without validating the applicability of our designs on mainstream generative models. Moreover, our model, tailored for sentence-level EAE tasks, has not been evaluated for document-level performance. Yet, experiments might be conducted with the help of proper segmentation of document-level records. Furthermore, the need for datasets with entity annotations places a high demand on dataset quality. Our current sequence labeling method is not suitable for cases of entity overlap, potentially resulting in reduced effectiveness when a contextual span encompasses multiple entity types.

Ethics Statement

In our research and experimental process of event argument extraction, we utilize pre-trained models trained on specific corpus and publicly available datasets required for the task. To our knowledge, the aforementioned pre-trained models and datasets have been widely applied to relevant missions for many years without apparent ethical issues. Meanwhile, we keep honest in our work and our work will not be used to harm anyone.

Acknowledgments

This work was supported by Beijing Natural Science Foundation (Grant No.4222032) and the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No.61921003).

References

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

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

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), pages 4380–4397, Toronto, Canada. Association for Computational Linguistics.

Yong Guan, Jiaoyan Chen, Freddy Lecue, Jeff Pan, Juanzi Li, and Ru Li. 2023. Trigger-argument based explanation for event detection. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5046–5058, Toronto, Canada. Association for Computational Linguistics.

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

Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415.

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

John D. Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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

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

Jian Liu, Chen Liang, Jinan Xu, Haoyan Liu, and Zhe Zhao. 2023a. Document-level event argument extraction with a chain reasoning paradigm. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9570–9583, Toronto, Canada. Association for Computational Linguistics.

Jian Liu, Dianbo Sui, Kang Liu, Haoyan Liu, and Zhe Zhao. 2023b. Learning with partial annotations for event detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 508–523, Toronto, Canada. 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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. Cite arxiv:1711.05101Comment: Published as a conference paper at ICLR 2019.

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.

Mingyu Derek Ma, Alexander Taylor, Wei Wang, and Nanyun Peng. 2023. DICE: Data-efficient clinical event extraction with generative models. In *Proceedings of the 61st Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 15898–15917, Toronto, Canada. Association for Computational Linguistics.

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

Minh Van Nguyen, Viet Dac Lai, and Thien Huu Nguyen. 2021. Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 27–38, Online. Association for Computational Linguistics.

Hao Peng, Xiaozhi Wang, Feng Yao, Kaisheng Zeng, Lei Hou, Juanzi Li, Zhiyuan Liu, and Weixing Shen. 2023. The devil is in the details: On the pitfalls of event extraction evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9206–9227, Toronto, Canada. Association for Computational Linguistics.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ERE: Annotation of entities, relations, and events. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98, Denver, Colorado. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, page 5998–6008. Curran Associates, Inc.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, 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.

Xingyao Wang, Sha Li, and Heng Ji. 2023. Code4Struct: Code generation for few-shot event structure prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3640–3663, Toronto, Canada. Association for Computational Linguistics.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 2021. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3533–3546, Online. Association for Computational Linguistics.

Zhiyang Xu, Jay Yoon Lee, and Lifu Huang. 2023. Learning from a friend: Improving event extraction via self-training with feedback from Abstract Meaning Representation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10421–10437, Toronto, Canada. Association for Computational Linguistics.

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

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.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 337–346, Hong Kong, China. Association for Computational Linguistics.

Tong Zhu, Xiaoye Qu, Wenliang Chen, Zhefeng Wang, Baoxing Huai, Nicholas Yuan, and Min Zhang. 2022. Efficient document-level event extraction via pseudo-trigger-aware pruned complete graph. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4552–4558. International Joint Conferences on Artificial Intelligence Organization. Main Track.

A Data and Training Details

A.1 Templates

Table 7 presents template designs for several events in the ACE05-E, ACE05-E$^+$, and ERE datasets. The trigger words in these templates are related to context and temporarily denoted by "xxx."

A.2 Role-Entity Mapping

Table 9 displays the correspondence between argument roles and entity types for certain events. This correspondence is one-to-many and derived from annotations in datasets.

A.3 Datasets

Table 8 demonstrates the splits of the training, validation, and test sets for the three datasets, along with the statistics of sentences, entities, and arguments for each split.

A.4 Hyperparameters

Table 10 shows the hyperparameter settings for Scented-EAE on three benchmarks. The selection of the optimal thresholds is determined through numerous experimental attempts. In addition, our code also facilitates the evaluation of multiple sets of thresholds simultaneously.

A.5 LLM Results

Table 11 presents the experimental results of applying our design to the LLMs (T5-3B, LLAMA3-8B).

Dataset	Event Type	Template
ACE05-E	Movement.Transport	In the Movement.Transport event triggered by xxx, Agent (People or Organization) transported Artifact (Vehicle or Facility) in Vehicle (Vehicle or Facility) cost Price from Origin (Location) place to Destination (Location) place.
	Personnel.Elect	In the Personnel.Elect event triggered by xxx, Entity (People or Organization) elected Person (People or Organization) at Place (Location) for Position.
	Contact.Phone-Write	In the Contact.Phone-Write event triggered by xxx, Entity (People or Organization) communicated remotely at Place (Location).
ACE05-E$^+$	Life.Die	In the Life.Die event triggered by xxx, Agent (People or Organization) killed Victim (People) with Instrument (Vehicle or Weapon) at Place (Location).
	Business.Merge-Org	In the Business.Merge-Org event triggered by xxx, Org (Organization) merged at Place .
	Justice.Convict	In the Justice.Convict event triggered by xxx, Adjudicator (People or Organization) court or judge convicted Defendant (People or Organization) at Place (Location) for Crime.
ERE	Contact.Broadcast	In the Contact.Broadcast event triggered by xxx, Entity (People or Organization) made announcement to Audience (People or Organization) at Place (Location).
	Manufacture.Artifact	In the Manufacture.Artifact event triggered by xxx, Agent (People or Organization) developed Artifact (Vehicle or Facility) at Place (Location).
	Conflict.Demonstrate	In the Conflict.Demonstrate event triggered by xxx, Attacker (People or Organization) attacked Target (Object) hurting Victim using Instrument (Vehicle or Weapon) at Place (Location).

Table 7: Some examples of templates in three datasets.

Dataset	Split	#Sents	#Entities	#Args
ACE05-E	Train	17,172	20,006	4,859
	Dev	923	2,451	605
	Test	832	3,017	576
ACE05-E$^+$	Train	19,216	47,554	6,607
	Dev	901	3,423	759
	Test	676	3,673	689
ERE	Train	8,886	22,831	4,372
	Dev	720	1,949	378
	Test	604	1,621	257

Table 8: Statistics of datasets.

Dataset	Event Type	Role	Entity Types
ACE05-E	Business.Start-Org	Agent	People or Organization
		Org	People or Organization
		Place	Location
	Life.Marry	Person	People or Organization
		Place	Location
	Transaction.Transfer-Ownership	Artifact	Vehicle or Facility or Organization
		Seller	People or Organization
		Beneficiary	People or Organization
		Buyer	People or Organization
		Place	Location
ACE05-E$^+$	Contact.Meet	Entity	People or Organization
		Place	Location
	Life.Injure	Victim	People
		Agent	People or Organization
		Place	Location
		Instrument	Vehicle or Weapon
	Business.Declare-Bankruptcy	Org	People or Organization
		Place	Location
ERE	Movement.Transport-Person	Agent	People or Organization
		Destination	Location
		Person	People
		Origin	Location
		Instrument	Vehicle
	Contact.Correspondence	Entity	People or Organization
		Place	Location
	Personnel.Nominate	Agent	People or Organization
		Person	People
		Place	Location

Table 9: Some examples of role-entity correspondence in three datasets.

Hyperparameter	**Value**		
	ACE05-E	**ACE05-E$^+$**	**ERE**
Max Sequence Length	220	220	360
Bart Learning Rate	[1e-5,1e-4]	[1e-5,1e-4]	[1e-5,1e-4]
Other Learning Rate	[1e-5,1e-4]	[1e-5,1e-4]	[1e-5,1e-4]
Warm Up Ratio	0.1	0.1	0.1
Batch Size	16	16	8
Epoch	30(base)/50(large)	30(base)/50(large)	30(base)/50(large)
Maximum Gradient Norm	5	5	5
Weight Decay	0.01	0.01	0.01
Best Start Threshold	0.9(base)/0.50(large)	0.65(base)/0.9(large)	0.7(base)/0.35(large)
Best End Threshold	0.15(base)/0.85(large)	0.25(base)/0.6(large)	0.35(base)/0.2(large)

Table 10: Hyperparameters of training Scented-EAE.

LLM	**ACE05-E**	**ACE05-E$^+$**	**ERE**
T5-3B	72.5	71.9	70.7
LLAMA3-8B	62.4	61.9	60.6

Table 11: Arg-C F1 of LLMs.