TMAP: Type-Aware Multi-AMR Prompt for Event Argument Extraction

Anonymous ACL submission

Abstract

Event Argument Extraction (EAE) is a crucial task for identifying the specific roles of arguments within events. Current generation-based EAE models have demonstrated significant improvements in performance. However, these methods often overlook the semantic structural associations between other related event types. To fill this gap, we propose a novel method called Type-aware Multi-AMR Prompt (**TMAP**) for event argument extraction, which integrates multiple related event-type semantic structural information. Specifically, we use AMR graphs to capture the semantic structure information of input passages. We also integrate the multiple AMR graphs from other passages that belong to the same event type and similar event type to enhance the semantic information of the input passages' AMR graph. Additionally, we construct three different prompts based on multiple AMR graphs to help the model extract event argument more effectively. We conduct extensive experiments and analyses on RAMS and WIKIEVENTS datasets to show that TMAP achieves significant improvements in F1 score, demonstrating its efficiency in EAE tasks.

1 Introduction

Event Argument Extraction (EAE) is a crucial task in Natural Language Processing (NLP) that aims to identify event-related arguments and categorize them into specific roles in a passage (Li et al., 2022). As illustrated in Figure 1, given a passage, and the event trigger is *appeal*, the goal of EAE is to extract the arguments associated with the *appeal* event. These arguments include *districts*(role=Plaintiff), *U.S. Supreme Court* (role=Adjudicator), *Washington* (role=Place), with the argument roles indicated in parentheses. EAE plays an essential role in various downstream applications, including Knowledge Graph Construction (Zhong et al., 2023), Recommendation Systems (Li et al., 2020b), and Dialogue Systems (Zhang et al., 2020).

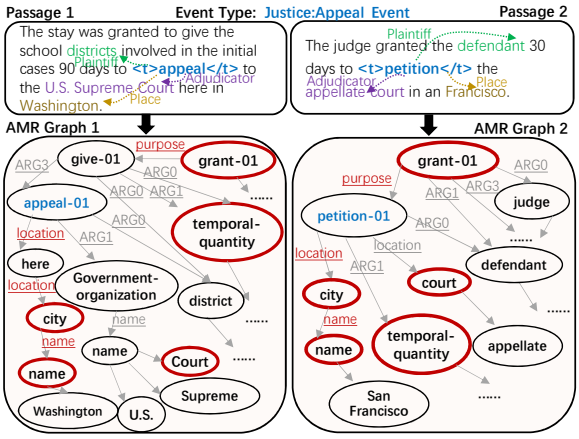

Figure 1: This figure illustrates two passages with the same event type and their corresponding argument information. Their AMR graphs show a high degree of similarity in both structure and content. The text highlighted in blue denotes the event triggers, while the green, purple, and yellow represent different arguments and their respective roles. The red elements indicate the shared information between the two AMR graphs.

Early research on EAE focused on formulating EAE as classification or sequence labeling problems (Chen et al., 2015; Huang and Jia, 2021; Li et al., 2022). Recently, there has been increasing attention on generation-based methods (Ma et al., 2022; Hsu et al., 2023, 2022; Xu et al., 2023) for tackling EAE, which have demonstrating significant performance improvements utilizing of the Pre-trained Language Models (PLMs) (Lewis et al., 2020). For instance, PAIE (Ma et al., 2022) and DEGREE (Hsu et al., 2022) introduced a generation-based method generating prompts/templates that contained argument role information for each event type and efficient extraction of all arguments. These methods ignore abundant semantic structure and linguistic information in input passages. AMPERE (Hsu et al., 2023) proposed a method to incorporate auxiliary semantic structure and syntac-

tic information into generation-based EAE models. This method utilizes Abstract Meaning Representation (AMR) (Banarescu et al., 2013) to summarize the semantic structure information of the input passages and generate a graph. Many of the nodes and edges in the AMR graph share strong similarities with the event structures. However, AMPERE mainly focused on receiving input passage AMR information, regardless of the semantic structures correlation between other related event types.

In Figure 1, there are two AMR graphs of two different passages belonging to a similar event type that share many similar semantic structures. (Liu et al., 2022) has emphasized the importance of utilizing the semantic information of related event types from training data to extract valuable information for input passage and event type. It is intuitive that passages of the related event types share similar information, such as text structure and argument role, etc. This information can be leveraged to enhance the AMR graph information of the input passage, thus improving the model's ability to capture semantic information and enhancing the performance of EAE.

In this paper, we propose a novel generation-based method called **T**ype-aware **M**ulti-**A**MR **P**rompt (**TMAP**) for event argument extraction (EAE). TMAP integrates multiple related event-type AMR graphs and constructs various prompts to enhance the performance of EAE. Specifically, to enhance the semantic information of the AMR graph for the input passage, we integrate the multiple AMR graph information of the passage related to the same event type and similar event types. Additionally, we construct three different prompts based on multiple AMR graphs, including AMR prompt, related event type prompt, and related event type AMR prompt. This assists the model in extracting event arguments more effectively.

Our contributions can be summarized as follows:

- We propose TMAP, a generation model based on AMR that enhances the model's capability to capture the semantic structure and improve the performance of EAE by incorporating multiple AMR information from input passage as well as from other passages related to the same event type and similar event types.
- We generate three different prompts using multiple AMR graph information and the related event-type information to enrich semantic information to tackle the EAE tasks.
- We conducted extensive experiments on widely used datasets. The results demonstrate that our model outperforms the baseline methods using both the BART-base and BART-large model on these benchmarks. The ablation study further illustrates the effectiveness of key components in our approach.

2 Related Works

Event Argument Extraction (EAE) is a complex and crucial task in the domain of natural language processing (NLP), garnering significant attention from researchers in recent years (Li et al., 2022; Xu et al., 2023). Existing research on EAE can broadly be divided into two main paradigms. The first is the traditional approach, where the EAE task is typically framed as either a classification problem (Chen et al., 2015) or a sequence labeling task (Wei et al., 2021). The second, more recent approach, explores generative methods for EAE, where the task is reformulated as either question-answering (QA)-based (Liu et al., 2020) or generation-based (Ma et al., 2022; Xu et al., 2023).

2.1 Classification-based EAE

In classification-based Event Argument Extraction (EAE), the task is often framed as either a semantic role labeling problem (Wei et al., 2021) or as a textual entailment (White et al., 2017; Lyu et al., 2021; Sainz et al., 2022) and Question Answering (QA) task (Lyu et al., 2021; Du and Cardie, 2020; Li et al., 2020a; Liu et al., 2020; Wei et al., 2021; Liu et al., 2021). Semantic role labeling generally involves identifying spans and classifying their roles, but it can suffer from error propagation due to reliance on candidate selection. The textual entailment approach rephrases EAE by generating hypotheses based on predefined templates, then applies an entailment model to classify argument roles.

QA-based models are adept at recognizing argument boundaries using role-specific questions but require multiple questions per sentence, making the process inefficient. In contrast, our work treats event extraction as a condition-generation task, offering greater flexibility and reducing the annotation complexity of traditional methods.

2.2 Generation-based EAE

Recent advances in event extraction have shifted towards treating the task as a sequence genera-

tion problem. This approach includes transforming event extraction into tasks like structured translation (Paolini et al., 2021), constrained decoding (Lu et al., 2021), and template-based conditional generation (Li et al., 2021a; Hsu et al., 2022). Among these, the closest to our work are methods that rely on manually crafted discrete templates (Li et al., 2021a; Hsu et al., 2022), but these approaches often suffer from sub-optimal performance due to the rigidity of the templates. Few studies have explored methods for integrating auxiliary syntactic and semantic information into event extraction models. The AMPERE (Hsu et al., 2023) framework demonstrated the utility of Abstract Meaning Representation (AMR) for capturing these aspects.

However, AMPERE only leverages AMR information from individual passages, overlooking potential information sharing between different passages and limiting their ability to capture richer semantic dependencies. To address these limitations, we fill this gap by incorporating both AMR and dynamic event types' information as interactive prompts to provide enriched contextual knowledge for event argument extraction.

3 Method

We propose using a multiple Abstract Meaning Representation (AMR) graph to tackle the event argument extraction tasks. As shown in Figure 2, our model first extracts the dynamic type information related to the input passage. Next, we utilize the input passage and the dynamic type information to derive the multiple AMR graph. These graphs are then transformed into dense vectors through AMR linearization(Banarescu et al., 2013). Finally, we construct three different prompts based on the dynamic type information and the multiple AMR graph.

3.1 Problem Statement

Event Argument Extraction (EAE) is the task of identifying and categorizing event arguments into specific roles in the input passage. Following the framework of (Ma et al., 2022), we employ the BART (Lewis, 2019) model, a transformer-based pre-trained language model (PLM) consisting of an encoder-decoder architecture, $L = [L_{enc}, L_{dec}]$, and formulate the EAE task as a sequence generation problem.

Given an instance $(P_u, t, e_v, R(e)) \in D$, where P_u represents the passage in the passage set $\mathcal{P}$, $t \in P_u$ is the trigger word, e_v denotes the event type in the event type set $\mathcal{E}$, $R(e)$ is the set of argument roles, and D is the dataset, the goal is to extract a set of arguments $\{A(r)\}$, where each $A(r)$ corresponds to a span within P_u, and identify the argument role $r \in R(e)$.

3.2 Dynamic Type Information Generation

Recent EAE methods have predominantly focused on extracting information from the same event type, often neglecting the potential value of associations between similar event types. (Liu et al., 2022) proposed leveraging related event types to provide additional, complementary information. To effectively capture similar event type information for each passage, we compute the relationship between passages and event types using similarity scores.

First, we use the BART encoder to encode both the passage $\mathcal{P}$ and the event type $\mathcal{E}$, obtaining token-level representations by averaging the hidden vectors from the final encoder layer. We then concatenate these representations and re-encode the concatenated vector using the BART encoder to compute attention-based similarity scores, denoted as $\text{Sim}(P_u, e_v)$. These similarity scores quantify the degree of relevance between the passage and the event type.

Leveraging these similarity scores, we derive dynamic event type information for each passage. To organize this information and capture the relationships between passages and event types, we construct a graph $\mathcal{G} = \{V, E, W\}$, where V represents the node set consisting of both passages and event types, E denotes the edge set, and W represents the edge weights. The edge weights range from $[0, 1]$, reflecting the strength of the relationship between a passage and an event type.

Specifically, if a passage P_u belongs to a particular event type e_v, we create an edge between the corresponding passage and event type nodes, assigning a weight of 1. Additionally, for each passage, we connect it to the top-k most similar event types based on similarity scores, where the edge weight corresponds to the similarity score $\text{Sim}(P_u, e_v)$:

$$w_{u,v} = \begin{cases} 1 & \text{if } P_u \in e_v \\ \text{Sim}(P_u, e_v) & \text{if } P_u \in \text{top-}k \text{ similar } e_v \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

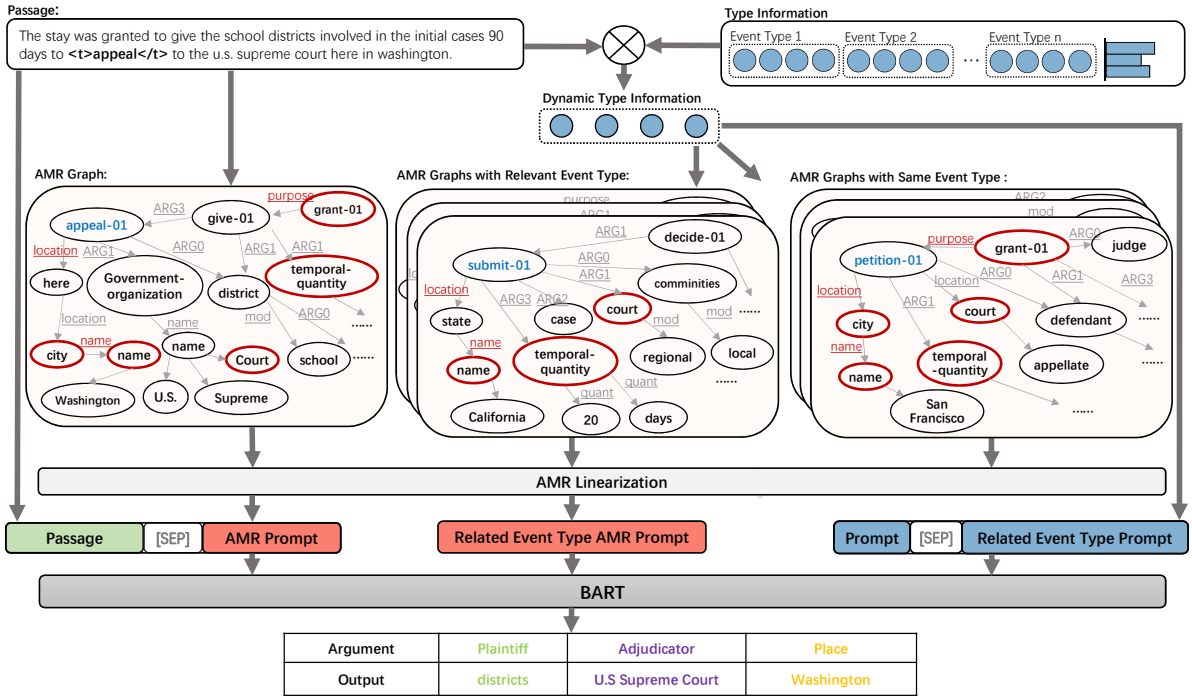

Figure 2: The main framework of our TMAP. Given a passage and an event trigger, we first generate dynamic-type information. Next, we utilize the input sentence and the dynamic type information to derive the AMR graph of the input sentence and its related AMR graph. These graphs are then transformed into dense vectors through AMR linearization. Finally, we generate three different prompts based on the dynamic type information and AMR graph.

This approach enriches the graph $\mathcal{G}$ by incorporating edges between passages and similar event types, allowing the model to leverage event-related context across passages for improved event argument extraction.

3.3 AMR Parsing and Linearization

As illustrated in Figure 1, Abstract Meaning Representation (AMR) (Banarescu et al., 2013) is a structured semantic framework designed to represent the meaning of English sentences as structured, directed graphs, where nodes represent entities or concepts and edges describe the relationships between these concepts.

In this paper, we adopt the SPRING (Bevilacqua et al., 2021) AMR parser, a BART-based model trained on the AMR 3.0 dataset[1], following the approach of AMPERE (Hsu et al., 2023). SPRING generates the AMR graph A_{P_u} for each passage. The AMR graph is then employed as additional input to the BART to enrich the passage representation.

To maximize the utilization of dynamic event type information from graph $\mathcal{G}$ we created in 3.2, and we combined it with our AMR graph. Specifically, for each passage P_u, we obtain a set of passages $\mathcal{P}_\mathcal{S}$ that share the same event type set and a set $\mathcal{P}_\mathcal{R}$ with similar event types. For each passage $P_{S_i} \in \mathcal{P}_\mathcal{S}$ and $P_{R_j} \in \mathcal{P}_\mathcal{R}$, we construct the corresponding AMR context sets $\mathcal{A}_\mathcal{S}$ and $\mathcal{A}_\mathcal{R}$, respectively.

To encode these AMR graphs, we follow the approach of (Konstas et al., 2017) by adopting a depth-first-search (DFS) algorithm to linearize the AMR graph into a sequence. And to verbs such as *appeal-01* are simplified to their root form, *appeal*, to remove extraneous information. As shown in Table 1 of (Zhang and Ji, 2021), we replace categor-

Categories	AMR Relation Types
Spatial	location, destination, path
Temporal	year, time, duration, decade, weekday
Means	instrument, manner, topic, medium
Modifiers	mod, poss
Operators	op-X
Prepositions	prep-X
Core Roles	ARG0, ARG1, ARG2, ARG3, ARG4
Others	Other AMR relation types

Table 1: Manually defined AMR relation clusters

[1] https://catalog.ldc.upenn.edu/LDC2020T02

ical semantic relationships with their corresponding categories to enhance the model's understanding of textual semantics. Based on predefined window size, we trim the resulting linearized sequence to obtain the necessary AMR context $\tilde{A_{P_u}}$, shared event type AMR context set $\tilde{\mathcal{A}_{\mathcal{P}_\mathcal{S}}}$ and similar event types AMR context $\tilde{\mathcal{A}_{\mathcal{P}_\mathcal{R}}}$, serving for further AMR prompt construction.

3.4 Type-Aware Prompt Generation

In this paper, we create three different prompts for event argument extraction, these prompt including AMR Prompt, Related Event Type AMR Prompt, and Related Event Type Prompt.

AMR Prompt and Related Event Type AMR Prompt Generation: After getting $\tilde{A_{P_u}}$ that orientated to passage P_u, we intend to combine it with the passage processed according to the method proposed by (Ma et al., 2022). Specifically, for a given passage P_u, we inject the trigger markers $<t>$ and $</t>$ before and after the trigger word t in P_u:

$$\tilde{P_u} = [x_1, x_2, \ldots, \langle \mathbf{t} \rangle, t, \langle / \mathbf{t} \rangle, \ldots, x_n] \quad (2)$$

Then we combined the passage and its processed AMR context into the BART encoder to return a sequence of representations $H_{P_u}^{(\text{enc})}$. The output of the encoder is formalized as:

$$H_{P_u}^{(\text{enc})} = L_{\text{enc}}\left(\tilde{P_u}; \tilde{A_{P_u}}\right) \quad (3)$$

This approach ensures that both the syntactic information from the passage and the structural semantics from the AMR graph are jointly encoded, providing a more comprehensive representation for downstream tasks.

As for passages set with the same event type $\tilde{\mathcal{A}_{\mathcal{P}_\mathcal{S}}}$ and similar event types $\tilde{\mathcal{A}_{\mathcal{P}_\mathcal{R}}}$, we utilize Sentence-BERT (Reimers and Gurevych, 2019) L_{ST} as the sentence transformer model to compute token-level representations separately. We then apply average pooling to compute the final AMR representations for the passages. The resulting embeddings are used to represent both passages with the same event type and passages with similar event types. These representations are formalized as follows:

$$\begin{aligned} H_{P_u}^{same} &= \frac{1}{|\mathcal{P}_\mathcal{S}|} \sum_{\tilde{A_j} \in \tilde{\mathcal{A}_{\mathcal{P}_\mathcal{S}}}} L_{ST}\left(\tilde{A_j}\right) \\ H_{P_u}^{rel} &= \frac{1}{|\mathcal{P}_\mathcal{R}|} \sum_{\tilde{A_k} \in \tilde{\mathcal{A}_{\mathcal{P}_\mathcal{R}}}} L_{ST}\left(\tilde{A_k}\right) \end{aligned} \quad (4)$$

The final related event type AMR prompt is constructed by concatenating the pooled representations from the same and similar event type passages:

$$H_{A_pt} = H_{P_u}^{same} \oplus H_{P_u}^{rel} \quad (5)$$

where H_{A_pt} represents the passage-oriented AMR embedding that incorporates both identical and related event-type information. This related AMR prompt is subsequently fed into the BART encoder to generate enriched representations that are both event-specific and context-aware.

$$H_{A_pt}^{(enc)} = L_{enc}\left(H_{A_pt}\right) \quad (6)$$

This approach enhances the model's ability to incorporate structured semantic knowledge from related passages, enabling a more contextually informed representation of the input passage.

Related Event Type Prompt Generation: Having gotten the dynamic information we introduced in section 3.2, we can leverage information from graph $\mathcal{G}$ to retrieve passages with identical or similar event types.

To compute the representations of nodes in $\mathcal{G}$, we employ a Graph Attention Network (GAT) (Veličković et al., 2018). After L layers of transformation, we obtain a final representation for each node. The feature vectors fed into the GAT are initialized using the passage and event type representations, which are generated by a Sentence-BERT L_{ST} to produce semantically rich and context-aware embeddings for both passages and event types. These embeddings serve as input to the GAT, ensuring that the node representations capture the contextual relationships between passages and event types.

Subsequently, the passage representation is concatenated with its corresponding event type representation, as well as the top-k similar event type representations, to form a related prompt, denoted as Pt_{rel} for passage P_u. This related prompt is combined with event-type-specific prompts Pt, following the method outlined by (Ma et al., 2022), to form a final composite prompt:

$$H_{pt} = (Pt; Pt_{rel}) \quad (7)$$

This formulation enhances the passage representations with event type information and related contextual details, enabling more informed downstream tasks by capturing the structural relationships in $\mathcal{G}$.

3.5 Training

Having gathered all the necessary prompts, our final task is to use this information to extract the event arguments. As introduced above, we had AMR prompt representation $H^{(enc)}_{A_pt}$, related event type AMR prompt combined with passage $H^{(\text{enc})}_{P_u}$ representation, and related event type prompt representation H_{pt}. In the next step, the BART decoder is employed to learn richer representations for the context and prompt using their interactions via cross-attention in multiple layers, returning the representations:

$$Output = L_{dec}\left(H^{(\text{enc})}_{P_u}; H^{(enc)}_{A_pt}; H_{pt}\right) \quad (8)$$

Here, we employ two selection heads and $Output$ to compute start and end span selectors represented as $\phi^{e,\text{start}}_j$ and $\phi^{e,\text{end}}_j$. The golden span for this role is denoted by $(a^{t,\text{start}}_j, a^{t,\text{end}}_j)$. It will be set to $(0, 0)$ if event e_k does not have an argument of this role in P_u. As such, the extractive prompt framework is utilized to estimate distributions over token positions in $\mathcal{P}$ for how likely each token in P_u would serve as the start/end position for the argument span of the role: $P^{e,\text{start}}_j = \text{softmax}(\phi^{e,\text{start}}_j Output)$, $P^{e,\text{end}}_i = \text{softmax}(\phi^{e,\text{end}}_j Output)$. Finally, to train our model, we optimize the loss function:

$$\mathcal{L} = -\sum_{e_k \in P}\sum_{j}(\log(P^{e,\text{start}}_j(a^{e,\text{start}}_j)) + \log(P^{e,\text{end}}_j(a^{e,\text{end}}_j))) \quad (9)$$

4 Experiment

4.1 Dataset

We conduct comprehensive experiments on two widely recognized EAE benchmark datasets: RAMS (Ebner et al., 2020) and WIKIEVENTS (Li et al., 2021b), which have been extensively utilized in previous studies (Ma et al., 2022). The RAMS dataset comprises 3,993 paragraphs, annotated with 139 event types and 65 argument roles.

DataSet	Split	Doc	Event	Argument
RAMS	Train	3,194	7,394	17,026
	Dev	399	924	2,188
	Test	400	871	2,023
WIKIEVENT	Train	206	3,241	4,542
	Dev	20	345	428
	Test	20	365	556

Table 2: Data statistics of RAMS and WIKIEVENT.

The WIKIEVENTS dataset consists of 246 passages that have been annotated with 50 event types and 59 argument roles.

4.2 Evaluation Metrics

In the experiment, We mainly used the following evaluation metrics:

- Argument Identification F1 score (**Arg-I**) denotes that an event argument is correctly identified if its offsets and event type match those of any of the argument mentions.
- Argument Classification F1 score (**Arg-C**) denotes that an event argument is correctly classified if its role type is also correct.
- Argument Head F1 score (**Head-C**) We also additionally evaluate Head-C for WIKIEVENTS, which only concerns the matching of the headword of an argument.

4.3 Baselines

In our comparison of TMAP with two groups of EAE work, the first group consists of classification-based EAE methods.

- **EEQA** (Du and Cardie, 2020): a model that employs a QA approach, utilizing multiple rounds of separated QA pairs to perform sentence-level EAE tasks.
- **FEAE** (Wei et al., 2021): a classification-based method for D-EAE by considering the interactions of event arguments through knowledge distillation.
- **DocMRC** (Liu et al., 2021): a D-EAE model incorporating both implicit knowledge transfer and explicit data augmentation regimes.

Model	PLM	RAMS		WIKIEVENTS		
		Arg-I	Arg-C	Arg-I	Arg-C	Head-C
EEQA (Du and Cardie, 2020)	BERT-b	46.4	44.0	54.3	53.2	56.9
FEAE (Wei et al., 2021)	BERT-b	53.5	47.4	-	-	
DocMRC (Liu et al., 2021)	BERT-b	-	45.7	-	43.3	
BART-Gen (Li et al., 2021a)	BART-b	50.9	44.9	47.5	44.7	44.2
EAE2E (Zeng et al., 2022)	BART-b	-	-	64.5	58.6	61.7
EEQA-BART (PAIE implementation, 2022)	BART-b	49.4	46.3	60.3	57.1	61.4
PAIE (Ma et al., 2022)	BART-b	54.7	49.5	68.9	63.4	66.5
PGAD (Luo and Xu, 2023)	BART-b	53.7	45.9	69.6	64.1	66.0
EDGE (Li et al., 2023)	BART-b	55.2	49.7	68.2	62.8	65.9
TMAP(Ours)	**BART-b**	**56.0**	**50.3**	**71.0**	**65.8**	**69.0**
EEQA (Du and Cardie, 2020)	BERT-l	48.7	46.7	56.9	54.5	59.3
BART-Gen (Li et al., 2021a)	BART-l	51.2	47.1	66.8	62.4	65.4
EAE2E (Zeng et al., 2022)	BART-l	-	-	70.8	65.7	67.8
EEQA-BART (PAIE implementation, 2022)	BART-l	51.7	48.7	61.6	57.4	61.3
PAIE (Ma et al., 2022)	BART-l	56.8	52.2	70.5	65.3	68.4
APE(Single) (Zhang et al., 2023)	BART-l	56.3	51.7	70.6	65.8	68.4
TMAP(Ours)	**BART-l**	**57.6**	**53.2**	**72.3**	**67.0**	**69.4**

Table 3: Overall performance comparison with baseline models. PLM denotes the pre-trained language models utilized by each model. "b" in column PLM denotes the base model, and "l" denotes the large model. We highlight the best result and underline the second-best result, both the base model and large model, respectively.

- **EEQA-BART** (Ma et al., 2022): the pre-trained model of EEQA is replaced from BERT to BART, resulting in a new model called EEQA-BART.

The second group comprises generation-based EAE methods.

- **BART-GEN** (Li et al., 2021a): defines EAE as a sequence-to-sequence task and generates corresponding arguments in a predefined format a template-based conditional generation model that employs sequence-to-sequence modeling and prompt learning to generate arguments.

- **EAE2E** (Zeng et al., 2022): a template-based conditional generation model that employs sequence-to-sequence modeling and prompt learning to generate arguments.

- **PAIE** (Ma et al., 2022): this model utilizes an encoder-decoder EAE architecture in which the context and the specifically designed prompt for each event type are fed separately into the encoder and decoder to simultaneously extract event arguments.

- **PGAD** (Luo and Xu, 2023): a variant of the PGAD model that discards the diversity of sample components during inference and extracts event arguments utilizing solely one context-aware prompt representation.

- **EDGE** (Li et al., 2023): utilize intra-event and inter-event graphs to construct dependencies among roles across various scenarios.

- **APE(Single)** (Zhang et al., 2023): the model trained in the transfer-disable setting, which optimizes the Prefix only in the target dataset.

The performance of EAE is obtained by running the provided code over our preprocessed data using the same evaluation metrics for a fair comparison. The performance for other baselines is inherited from (Ma et al., 2022) and (Luo and Xu, 2023), which present the model PAIE with current best-reported results for RAMS and PGAD with best-reported results for WIKIEVNETS.

4.4 Implementation Details

We use a generative pre-trained BART-base and BART-large model as both the encoder and decoder. The batch size is set to 4 for the RAMS dataset and

16 for the WIKIEVENTS dataset, with a learning rate of $2e-5$ for both datasets. We optimize our model using the AdamW optimizer. The maximum event argument span length is set to 10, and the window size is set to 250. The encoder's maximum sequence length is set to 500 for both datasets, while the maximum sequence length of generated context-aware prompt representations is set to 50 for RAMS and 80 for the WIKIEVENTS dataset. We set the max length of AMR prompt representations as 200 for RAMS and 450 for WIKIEVENTS, and the lengths of the same AMR representations and relevant AMR representations for both datasets are both 50. Our experiments were conducted on an NVIDIA GTX 3090 GPU.

4.5 Overall Performance

In Table 3, we compare the overall results of our proposed approach with all baseline methods. Our findings that our approach, TMAP, achieves the highest F1 scores for Arg-I and Arg-C on both the RAMS and WIKIEVENTS datasets. Specifically, For the BART-base model, TMAP achieves an absolute Arg-C improvement of 0.6% and Arg-I improvement of 0.8% on RAMS. In addition, TMAP achieves an absolute Arg-C improvement of 1.7%, Arg-I improvement of 1.4%, and Head-C improvement of 2.5% on WIKIEVENTS. For the BART-large model, TMAP achieves an absolute Arg-C improvement of 1% and Arg-I improvement of 0.8% on RAMS. In addition, TMAP achieves an absolute Arg-C improvement of 1.2%, Arg-I improvement of 1.5%, and Head-C improvement of 1% on WIKIEVENTS. This demonstrates the excellent ability of TMAP to deal with EAE tasks.

Additionally, for all results, the score for Arg-I is strictly higher than Arg-C since Arg-I only requires the event argument span match while Arg-C also requires role match. For the WIKIEVENTS dataset, the score for Head-C is always higher than Arg-C. This is because Head-C focuses on matching the headword of an argument, while Arg-C evaluates all words.

4.6 Ablation study

In this section, we investigate the effectiveness of three different prompts by removing each prompt on the RAMS dataset in turn to reveal the contribution of the designed prompt in TMAP.

- **Related event type AMR Prompt (RAP)**. We drop out of the related event type AMR graph information.
- **AMR Prompt (AP)**. We additionally drop out of the AMR graph of the input passage.
- **Related event type Prompt (RP)**. We drop out of the related event type information in TMAP.

Model	PLM	Arg-I	Arg-C
TMAP (ours)	BART-b	**56.0**	**50.3**
-w/o RAP	BART-b	55.5	50.1
-w/o AP, RAP	BART-b	55.4	49.9
-w/o RP, AP, RAP	BART-b	54.7	49.5

Table 4: Ablation study on RAMS.

We summarize the experiment results of ablation studies in Table 4. We observe that the complete TMAP model achieves the best results, with scores of Arg-I 56.0% and Arg-C 50.3%, demonstrating the effectiveness of the overall model architecture. Removing prompts such as the related event type AMR prompt and AMR prompt leads to noticeable drops in performance, especially on the Arg-I metric. The most significant performance decrease occurs when all prompts (RAP, AP, RP) are removed, underscoring the importance of our prompt-based approach. This suggests that the inclusion of AMR prompts provides critical contextual information that enhances the model's ability to capture event arguments effectively.

5 Conclusion

In this paper, we propose a novel method called Type-aware Multi-AMR Prompt (TMAP) for event argument extraction, which integrates multiple AMR graph information and generates multiple prompts. Specifically, we utilize AMR graphs to capture the semantic structure information of input passages and integrate multiple AMR graphs from other passages belonging to the same event type and similar event types, enhancing the semantic information of the input passages' AMR graph. Using multiple AMR graphs, we construct three distinct prompts to help the model extract event argument more effectively. Experimental results demonstrate that the multiple AMR graphs module and multiple prompt modules significantly improve the performance of event argument extraction tasks.

Limitation

In our proposed TMAP model, we aim to improve the semantic structure information of the input passage AMR graph by integrating the AMR graph from other k passages that are associated with related event types. In our experiment, we set the hyperparameter $k = 3$, but we did not further investigate the impact of varying k on the experimental results. Additionally, another limitation is that we only tested TMAP on RAMS and WIKIEVENTS benchmarks. In our future work, we intend to assess our model on other EAE datasets, such as the ACE05 dataset.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12564–12573.

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.

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

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

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.

Yusheng Huang and Weijia Jia. 2021. Exploring sentence community for document-level event extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 340–351, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural amr: Sequence-to-sequence models for parsing and generation. *arXiv preprint arXiv:1704.08381*.

M Lewis. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

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.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020a. Event extraction as multi-turn question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 829–838.

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.

Manling Li, Alireza Zareian, Ying Lin, Xiaoman Pan, Spencer Whitehead, Brian Chen, Bo Wu, Heng Ji, Shih-Fu Chang, Clare Voss, Daniel Napierski, and Marjorie Freedman. 2020b. GAIA: A fine-grained multimedia knowledge extraction system. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 77–86, Online. Association for Computational Linguistics.

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

IEEE Transactions on Neural Networks and Learning Systems.

Sha Li, Heng Ji, and Jiawei Han. 2021a. 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.

Sha Li, Heng Ji, and Jiawei Han. 2021b. Document-level event argument extraction by conditional generation. *arXiv preprint arXiv:2104.05919*.

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

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

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.

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.

Lei Luo and Yajing Xu. 2023. Context-aware prompt for generation-based event argument extraction with diffusion models. *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332.

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. *arXiv preprint arXiv:2101.05779*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Oscar Sainz, Itziar Gonzalez-Dios, Oier Lopez de Lacalle, Bonan Min, and Eneko Agirre. 2022. Textual entailment for event argument extraction: Zero-and few-shot with multi-source learning. *arXiv preprint arXiv:2205.01376*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Kaiwen Wei, Xian Sun, Zequn Zhang, Jingyuan Zhang, Guo Zhi, and Li Jin. 2021. Trigger is not sufficient: Exploiting frame-aware knowledge for implicit event argument extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4672–4682, Online. Association for Computational Linguistics.

Aaron Steven White, Pushpendre Rastogi, Kevin Duh, and Benjamin Van Durme. 2017. Inference is everything: Recasting semantic resources into a unified evaluation framework. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 996–1005.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, and Enhong Chen. 2023. Large language models for generative information extraction: A survey. *arXiv preprint arXiv:2312.17617*.

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

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.

Tao Zhang, Min Chen, and Alex A. T. Bui. 2020. Diagnostic prediction with sequence-of-sets representation learning for clinical events. In *Artificial Intelligence in Medicine. Conference on Artificial Intelligence in Medicine (2005-)*, volume 12299, pages 348–358.

Zixuan Zhang and Heng Ji. 2021. Abstract meaning representation guided graph encoding and decoding for joint information extraction. In *Proc. The 2021 Conference of the North American Chapter of the Association for Computational Linguistics-Human Language Technologies (NAACL-HLT2021)*.

Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. 2023. A comprehensive survey on automatic knowledge graph construction. *ACM Computing Surveys*, 56(4):1–62.