Multiplex Graph Prompt Learning and Attentive Fusion for Event Graph Completion

Anonymous ACL submission

Abstract

This paper introduces an Event Graph Completion (EGC) task to predict the absence of multi-relations between events in a heterogeneous Event Graph (EG) with four prevailing relations. Considering that multi-relations may facilitate the completion of each other, that is, completing one type of event relations may benefit from the other type of relations. We propose a *Multiplex Graph **P**rompt **L**earning and **A**ttentive **F**usion* (PLAF) model to learn events' representations interactively among each homogeneous graphs and adopt a kind of dual graph prompt learning for missing relation prediction. The PLAF includes three modules: (1) *Dual Graph Prompt Learning* (DGPL) reformulates each EG into an event triplet sequence to encode its structural and semantic information; (2) *Multiplex Graph Attention Network* (MGAT) divides each heterogeneous EG into four homogeneous graphs to learn events' representations through both inter-graph and cross-graph attention; (3) *Aggregative Relation Prediction Module* (ARPM) aggregates the missing relation predictions from both DGPL and MGAT to complete the EGs. Extensive experiments on our constructed EGC-MAVEN dataset validate our arguments and show that our proposed PLAF outperforms the advanced competitors.

1 Introduction

Event Graphs (EGs) (Glavaš and Šnajder, 2015) is a structured representation of event knowledge that consists of events as nodes and relations between events as edges. It organizes and describes the development thread and evolution pattern of complex events in real-world scenarios, and significantly benefits tasks such as timeline summarization (Li et al., 2021b), event prediction (Ugai et al., 2024) and event schema induction (Jin et al., 2022). This paper introduces an Event Graph Completion (EGC) task that aims to complete the multi-relation

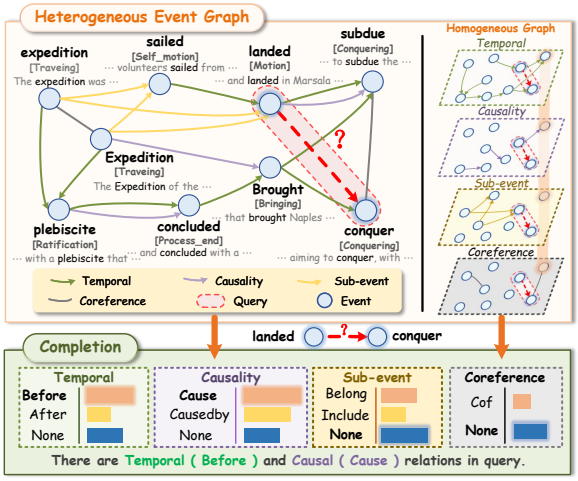

Figure 1: An example of event graph completion task.

edges between event nodes in a heterogeneous EG, so as to optimize the EG and acquire potential event relations.

As illustrated in Fig. 1, in an EG, each event node contains an event type, a mention word(s) and the raw sentence it belongs to; While the relation between events includes Temporal, Causality, Sub-event, and Coreference, in which the Coreference relation is undirectional and the other relations are directional. Note that, multi-relations may exist between two event nodes, as such the heterogeneous EG can be divided into four homogeneous graphs each within one type of event relation. Given an EG and an event pair in it (here the event *landed* and *conquer*), the EGC task is defined as completing the missing relations between the two given events, including determining whether one or more types of relations exist and if exist further identifying their directions (here the *Temporal (Before)* and the *Causality (Cause)* relations).

Existing graph completion tasks mainly focus on conventional knowledge graphs, which design various Graph Neural Network (GNN) to learn the representation of entity nodes for relation predictions (Yu et al., 2024; Lin et al., 2022; Wang et al.,

2022a). We argue that different types of event relations may facilitate the completion of other missing relations. For instance, the causality relations between events can help identify the temporal relations between them, as the causalities themselves are inherently sequential. Motivated from such considerations, we propose a multiplex graph attention network to learn events' representations interactively among different event relation types.

Besides, the recent Pre-trained Language Models (PLMs) exhibit outstanding ability in knowledge Q&A and logical reasoning, and have been applied in many downstream tasks (Liu et al., 2024; Zhu et al., 2024; Du et al., 2024). However, the PLM-based prompt learning commonly takes text sequences as input and struggles to be used in graph-structured tasks. To benefit from the abundant knowledge embedded in advanced PLMs, we adopt a kind of dual graph prompt learning paradigm to encode both the semantic and structure information of an EG for missing relation predictions.

In this paper, we introduce an EGC task to complete missing relations in EGs and propose a *Multiplex Graph **P**rompt **L**earning and **A**ttentive **F**usion* (PLAF) model for the EGC task. The PLAF model consists of three main modules: (1) *Dual Graph Prompt Learning* (DGPL) reformulates each EG into an event triplet sequence for both event mention and event type template construction, that are respectively input into a PLM to predict the missing relations within the prompt learning paradigm. (2) *Multiplex Graph Attention Network* (MGAT) divides each heterogeneous EG into four homogeneous graphs each with one relation type, and learns the representations of event node through both inter-graph and cross-graph attention. (3) *Aggregative Relation Prediction Module* (ARPM) aggregates the missing relation predictions from both DGPL and MGAT modules to complete the EGs.

We construct an EGC-MAVEN dataset based on the existing event relation corpus MAVEN-ERE (Wang et al., 2022b). Extensive experiments validate our arguments and show that our proposed PLAF outperforms the advanced competitors.

2 Related work

2.1 Knowledge Graph Completion

Knowledge Graph Completion (KGC) (Shen et al., 2022) aims to predict and infer the missing entity nodes or relation edges in knowledge graphs. Different from our proposed EGC task, it focuses on entity-centric knowledge graphs instead of event graphs.

The KGC can be divided into two subtasks: link prediction and triple classification. Link prediction (Jiang et al., 2016; Liu et al., 2022; Wei et al., 2024; Dimitriou and Karyotis, 2024) aims to predict the missing entities or relations in the triplets. For example, Ren et al. (2023) integrate a self-attention mechanism, positional encoding, and diachronic embedding technique to capture both graph structure and time series information for enhanced evaluation of a triplet. Wei et al. (2024) captures meta relational information from limited support instances to predict the missing entity. Triple classification (Ren et al., 2023; Wang et al., 2023; Shang et al., 2024) determines whether to add a triplet to KGs by estimating whether a triplet is true or not.

2.2 Event Graph Completion

Event prediction aims to predict the future event based on history events and relations between them, which is of great importance in many applications (Chaturvedi et al., 2017; Xiong et al., 2023; Li et al., 2024). For instance, Li et al. (2021a) utilize temporal complex event schemas, which are capable of representing multiple temporal dependencies between events and their connected arguments, to capture complex event structures and predict future events. Unlike the schema-based approaches, Islam et al. (2024) employ two view encoders to capture local dynamic structure and hierarchical dynamic graph to predict potential events for the next timestamp. Besides, Wang et al. (2022a) propose to complete event nodes in an event graph. They utilize the event schema template, which describes the stereotypical structure of event graphs, and the subgraph mapping algorithm for event node completion.

This paper aims to predict the absence of multi-relations between events in a heterogeneous event graph and propose a Multiplex Graph Prompt Learning and Attentive Fusion (PLAF) model for event graph completion.

3 Event Graph Completion

3.1 Task Definition

The *Event Graph Completion* (EGC) task is defined as completing the absence of multi-relations between two event nodes in a heterogeneous event graph, including Causality, Temporal, Subevent, and Coreference relations. As illustrated

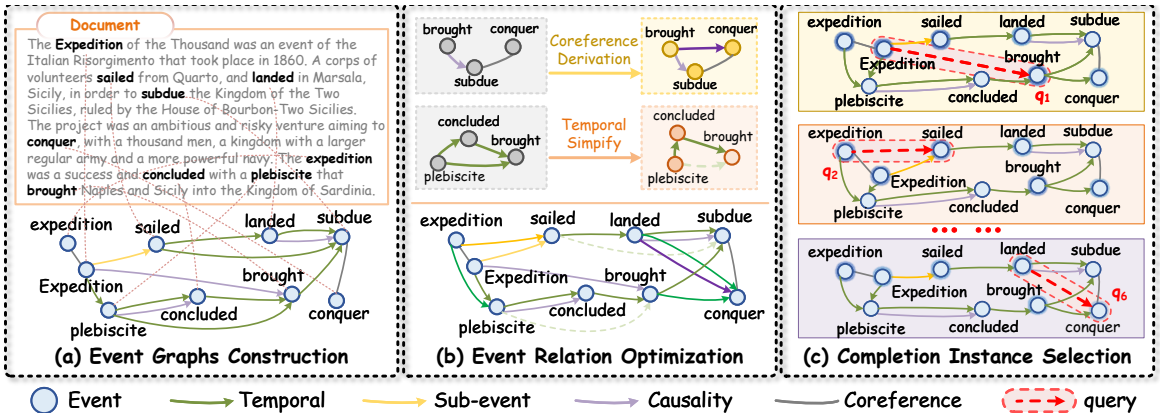

Figure 2: Illustration of our EGC-MAVEN dataset construction, including: (a) Event Graphs Construction; (b) Event Relation Optimization and (c) Completion Instance Selection.

in Fig. 1, the EG is a multi-relation directed graph consisting of events as nodes and four types of relations between events as edges, denoted by $\mathcal{G}(\mathcal{E}, \mathcal{R})$. Where, each event node $e_i \in \mathcal{E}$ contains an event mention word(s) Em_i and the raw sentence S_i it belong to, as well as its corresponding event type Et_i. The relation edge $r_{ij}^{\tau} \in \mathcal{R}$ is between two event nodes e_i and e_j, where $\tau \in \mathcal{S}$ is the relation type and $\mathcal{S} = \{$Temp., Caus., SubE., Corf.$\}$ We note that multiple types of relations may coexist between two event nodes, and except for the coreference relation which is unidirectional, the other three types of relations are directional. The objective of EGC task is to determine whether there exist any missing relations r_{ij}^{τ} between two given event nodes e_i and e_j (i.e. the query event pair q_{ij}) in an EKG $\mathcal{G}$, and if one or more relations exist, further identify their relation type(s) and direction(s).

3.2 Dataset Construction

We construct an EGC dataset, named EGC-MAVEN, based on the public event relation dataset MAVEN-ERE (Wang et al., 2022b), which has annotated event mentions and four types of event relations within documents. Fig. 2 illustrates the process of EGC dataset construction, including event graph construction, event relation optimization, and completion instance selection. [1]

• **Event graph construction:** We first construct EGs based on the annotations in each document from MAVEN-ERE dataset, using the annotated events as nodes and the four types of annotated event relations as edges. We remove very few EGs with temporal relation conflicts and causality relation cycles errors.

• **Event relation optimization:** Considering that events with coreference relations should have the same temporal, causality, and sub-event relations, we further derive more event relations based on the coreference events in each EG. For example, if the *subdue* event has a causality relation with the *brought* event, its coreference event *conquer* should also has a causality relation with the *brought* event. These derived relations are added to the constructed EGs. Besides, the fully annotated temporal relations in the MAVEN-ERE dataset make the EGs complex and redundant. To this end, we simplify the temporal relations in EGs to a kind of chain-like structure, such as the temporal relation chain {*plebiscite* → *concluded* → *brought*}, where a redundant temporal relation between {*plebiscite* → *brought*} is removed. More details about the relation optimization are in Appendix A.

• **Completion instance selection:** In each optimized EG, we randomly select six event pairs and mask all the relations between them to form incomplete EGs for completion, where the selected event pairs are the queries and the masked relations are to be predicted. Each EGC instance contains an incomplete EG $\mathcal{G}(\mathcal{E}, \mathcal{R})$, a query event pair q_{ij}, and a relation label set L in our EGC-MAVEN dataset. Where, the label set L may includes one or more types of event relations with their directions (if directional). To alleviate the data label imbalanced issues, we manually ensure that the six EGC instances from one EG must included all four types of event relations, as well as two none relation instances, after the randomly selection. Table 1 summarizes the statistics of our constructed EGC-MAVEN dataset.

[1]The EGC-MAVEN dataset will be publicly released after the anonymous review.

Dataset	Graph	Instance	Avg.Node	Avg.Edge	Sub-event		Temporal		Causality		Coreference
					Belong	Include	Before	After	Cause	Causedby	Cof
Train	2198	10991	26	60	567	557	1474	1478	1322	1220	1289
Dev.	708	3589	27	66	181	182	477	453	427	407	457
Test	708	3601	26	64	213	193	495	474	425	405	426
Total	3614	18181	——	——	961	932	2446	2405	2174	2032	2172

Table 1: Statistics of our EGC-MAVEN dataset.

4 Methodology

We propose a *Multiplex Graph Prompt Learning and Attentive Fusion* (PLAF) model for the EGC task, as illustrated in Fig. 3. The PLAF takes an event graph $\mathcal{G}(\mathcal{E}, \mathcal{R})$ and a query event pair q_{ij} as input to predict the absence of multi-relations between them. It contains three modules: (1) Dual Graph Prompt Learning (DGPL); (2) Multiplex Graph Attention Network (MGAT); (3) Aggregative Relation Prediction Module (ARPM).

4.1 Dual Graph Prompt Learning

To encode both the semantic and structure information of EGs, we first transform each EG into a kind of event triplet sequence, which enables us to construct two templates: event mention template and event type template. Then the two templates are respectively input into a PLM to encode the missing relation representations within the prompt learning paradigm.

(1) Graph Reformulation: We propose a Distance-First Linearization (DFL) approach to reformulate each input EG $\mathcal{G}(\mathcal{E}, \mathcal{R})$ into an event triplet sequence $\mathcal{C}(\mathcal{G})$. In this sequence, each event triplet $T_k = (e_m, r^{\tau}_{mn}, e_n)$ is composed of an event pair $e_m, e_n \in \mathcal{E}$ and the relation $r^{\tau}_{mn} \in \mathcal{R}$ between them, in the EG. The query triplet is denoted as $T_q = (e_i, r^{\tau}_q, e_j)$, where r^{τ}_q is the relation to be predicted. The event triplet sequence $\mathcal{C}(\mathcal{G})$ is the concatenation of the event triplets T_k and the query triplet T_q, represented as follows:

$$\mathcal{C}(\mathcal{G}) = T_1,\ T_2,\ \ldots,\ T_k,\ T_q. \tag{1}$$

We argue that the closer an event triplet T_k is to the query triplet T_q, the more critical information it can provide for its relation completion. To this end, our DFL approach orders event triplets based on their distances to the query triplet. The distance of an event triplet T_k to the query triplet T_q is computed by the number of edges on the shortest undirectional path from either one of its event nodes $e_a \in \{e_m, e_n\}$ to either one of the query's event nodes $e_b \in \{e_i, e_j\}$, as follows:

$$d_k(T_k, T_q) = \min_{p \in P(e_a, e_b)} |p|, \tag{2}$$

where $P(e_a, e_b)$ is the set of all undirected paths from the event triplet T_k to the query triplet T_q, $|p|$ is the number of edges on the path p. The event triplets are arranged in decreasing order of their distances to the query triplet, i.e. $d_1 \geq d_2 \geq \cdots \geq d_k$, indicating that T_k is the closest one to the query and T_1 is the farthest one.

(2) Template Construction: We respectively construct an event mention template $\mathcal{T}^m(\mathcal{C})$ and an event type template $\mathcal{T}^y(\mathcal{C})$ based on the event triplet sequence $\mathcal{C}(\mathcal{G})$, as the input to a PLM. In the event mention template $\mathcal{T}^m(\mathcal{C})$, each event triplet is formulated as a specific description of an event relation instance; While in the event type template $\mathcal{T}^y(\mathcal{C})$, each event triplet is formulated as a schema induction of event relation types, as follows:

$$\mathcal{T}^m_k = Em_m\ \ [Rel^{\tau}_{mn}]\ \ Em_n, \tag{3}$$
$$\mathcal{T}^y_k = Et_m\ \ [Rel^{\tau}_{mn}]\ \ Et_n, \tag{4}$$

where Em and Et are the event mention word(s) and event type word respectively, $[Rel^{\tau}_{mn}]$ is a relation word(s) selected from our predefined set $\mathcal{V}^{\tau}$ according to the ground truth event relations and directions, as follows:

$$\mathcal{V}^{SubE.} = \{Belong, Include, None\} \tag{5}$$
$$\mathcal{V}^{Caus.} = \{Cause, Causedby, None\} \tag{6}$$
$$\mathcal{V}^{Temp.} = \{Before, After, None\} \tag{7}$$
$$\mathcal{V}^{Corf.} = \{Cof, None\} \tag{8}$$

For the query triplet T_q, we apply similar templates to that of event triplets T_k, and use the PLM-specific token `[mask]` to represent relation word(s), which is next encoded to predict missing relations:

$$\mathcal{T}^m_q = Em_i\ \ \texttt{[mask]}_m\ \ Em_j, \tag{9}$$
$$\mathcal{T}^y_q = Et_i\ \ \texttt{[mask]}_y\ \ Et_j, \tag{10}$$

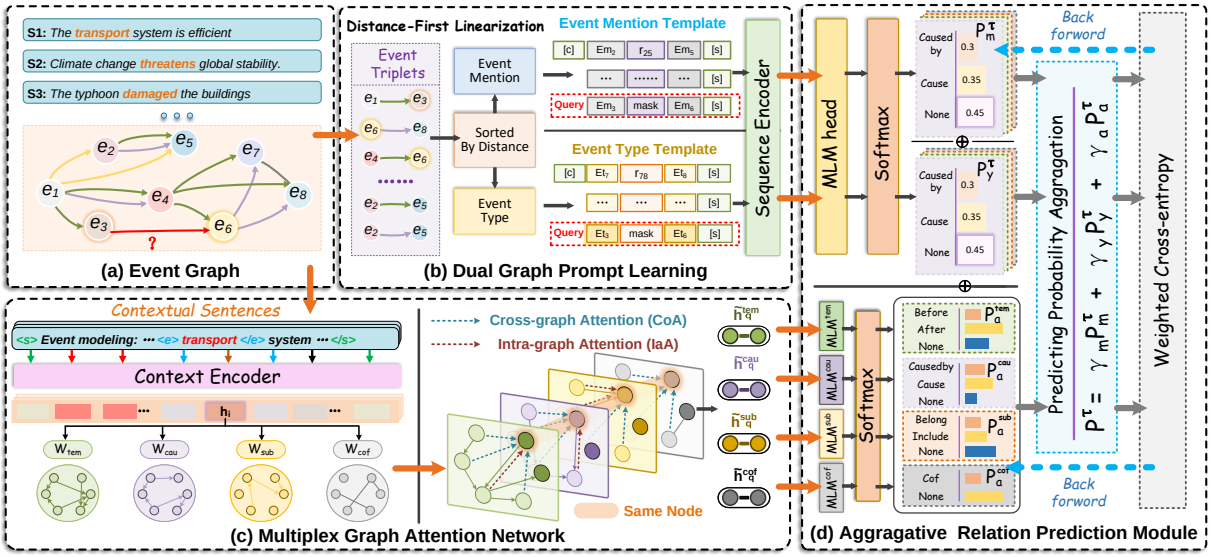

Figure 3: Illustration of our multiplex graph prompt learning and attentive fusion model.

Further, some PLM-specific tokens [CLS]([C]) and [SEP]([S]) are inserted into both templates to indicate its beginning and ending, as well as the boundary between each triplets. The event mention template $\mathcal{T}^m(\mathcal{C})$ and an event type template $\mathcal{T}^y(\mathcal{C})$ are represented as follows:

$$\mathcal{T}^m(\mathcal{C}) = \texttt{[C]}\ \mathcal{T}_1^m\ \texttt{[S]} \ldots \mathcal{T}_k^m\ \texttt{[S]}\ \mathcal{T}_q^m\ \texttt{[S]}, \quad (11)$$

$$\mathcal{T}^y(\mathcal{C}) = \texttt{[C]}\ \mathcal{T}_1^y\ \texttt{[S]} \ldots \mathcal{T}_k^y\ \texttt{[S]}\ \mathcal{T}_q^y\ \texttt{[S]}. \quad (12)$$

Note that, we set the distance between event nodes of the same event type to zero, in event type template construction. This results in a slightly different ordering of event triplets compared to that in the event mention template. More details can be found in Appendix B.

(3) PLM Encoding: We use a PLM named Sequence Encoder to encode each word in event mention template $\mathcal{T}^m(\mathcal{C})$ and event type template $\mathcal{T}^y(\mathcal{C})$, respectively. The hidden state vector of [mask] tokens from two template inputs $\boldsymbol{h}_{mask}^m$ and $\boldsymbol{h}_{mask}^t$ are used as the prompt representations of missing relations for next prediction.

4.2 Multiplex Graph Attention Network

We propose a MGAT module to model the interactions among different relation types, and learn the event node representations in an EG through both inter-graph and cross-graph attentions. The relation to be predicted is represented by the offset between two query event node representations.

First, we divide each heterogeneous event graph $\mathcal{G}$ with four kinds of relation edges into four homogeneous graphs $\mathcal{G}_\tau$ ($\tau \in \mathcal{S}$), each of which only has one type of relation edge. We initialize the representations of each event node $\boldsymbol{h}_i$ in an EG $\mathcal{G}$ by a PLM named Context Encoder. The contextual sentence S_i of each event is input to the PLM, and the output representation of event mention word(s) is used as the initialization representation of event node. They are next transformed by four learnable matrices $\boldsymbol{W}_\tau$ to obtain the event node representations for each homogeneous graphs $\boldsymbol{h}_i^\tau$, as follows:

$$\boldsymbol{h}_i^\tau = \boldsymbol{W}_\tau \boldsymbol{h}_i, \quad (13)$$

where $\tau \in \mathcal{S}$. Note that, each homogeneous graphs $\mathcal{G}_\tau$ has the same event nodes but with different representations. Further, we retain one type of event relation edges for each homogeneous graphs $\mathcal{G}_\tau$ and remove the other type of relations.

Then, we propose a multiplex graph attention mechanism to learn the representations of each event node $\tilde{\boldsymbol{h}}_i^\tau$, as follows:

$$\tilde{\boldsymbol{h}}_i^\tau = \omega Att(\boldsymbol{h}_i^\tau, \mathcal{N}_i^\tau) + \mu Att(\boldsymbol{h}_i^\tau, \mathcal{N}_i^{\mathcal{S}/\tau}), \quad (14)$$

where ω and μ are the weight coefficients of intra-graph and cross-graph attention respectively. The intra-graph neighbors $\mathcal{N}_i^\tau$ are the connected event nodes in the same homogeneous graphs $\mathcal{G}_\tau$; While the cross-graph neighbors $\mathcal{N}_i^{\mathcal{S}/\tau}$ are the connected event nodes in the other three homogeneous graphs $\mathcal{G}_{\mathcal{S}/\tau}$. The attention network $Att(\cdot)$ is computed as follows:

$$Att(\boldsymbol{h}_i, \mathcal{N}_i) = \sigma\left(\sum\nolimits_{j \in \mathcal{N}_i} s_{ij} \boldsymbol{W} \boldsymbol{h}_j\right), \quad (15)$$

where σ is the elu (Clevert, 2015) activation function, and s_{ij} is the attention coefficient, computed

as follows:

$$s_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\boldsymbol{a}^{\mathrm{T}}[\boldsymbol{W}\boldsymbol{h}_i \| \boldsymbol{W}\boldsymbol{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\boldsymbol{a}^{\mathrm{T}}[\boldsymbol{W}\boldsymbol{h}_i \| \boldsymbol{W}\boldsymbol{h}_k]\right)\right)} \tag{16}$$

where $\boldsymbol{a}$ and $\boldsymbol{W}$ are learnable parameters.

Finally, we learn different event node representations $\tilde{\boldsymbol{h}}_i^{\tau}$ for each homogeneous graph. The offsets between two query event node representations in each homogeneous graph are used as the feature representations of their corresponding relations to be predicted:

$$\tilde{\boldsymbol{h}}_q^{\tau} = \tilde{\boldsymbol{h}}_m^{\tau} - \tilde{\boldsymbol{h}}_n^{\tau}. \tag{17}$$

4.3 Aggregative Relation Prediction Module

The ARPM module aggregates the missing relation predictions from both DGPL and MGAT modules to complete the EG.

Specifically, we input the prompt representations of missing relations $\boldsymbol{h}_{mask}^{m}$ and $\boldsymbol{h}_{mask}^{y}$, obtained from DGPL, into a Masked Language Model (MLM) heads. They respectively estimate the probability of each word in its vocabulary $\mathcal{V}$ for the `[mask]` tokens. We apply a *softmax* on the predicted scores of our predefined relation word $\mathcal{V}^{\tau} \in \mathcal{V}$ to normalize them into relation probabilities, as follows:

$$P_m^{\tau} = \textit{Softmax}\left(\texttt{[mask]}_m = v | v \in \mathcal{V}^{\tau}\right), \tag{18}$$

$$P_y^{\tau} = \textit{Softmax}\left(\texttt{[mask]}_y = v | v \in \mathcal{V}^{\tau}\right), \tag{19}$$

where $\texttt{[mask]}_m$ and $\texttt{[mask]}_y$ are from event mention template $\mathcal{T}^m(\mathcal{C})$ and event type template $\mathcal{T}^y(\mathcal{C})$ respectively. Besides, We input the feature representation of missing relations $\tilde{\boldsymbol{h}}_q^{\tau}$, obtained from MGAT, into four Multi-Layer Perceptrons (MLPs) for relation prediction. They predict the probabilities P_a^{τ} of each relation type τ, as follows:

$$P_a^{\tau} = \textit{Softmax}\left(\textit{MLP}^{\tau}(\tilde{\boldsymbol{h}}_q^{\tau})\right), \tag{20}$$

where $\textit{MLP}^{\tau}(\cdot)$ is the MLP classifier for relation type τ.

We use weighted average to aggregate the relation predicting probabilities P_m^{τ}, P_y^{τ} and P_a^{τ} obtained from both DGPL and MGAT, as final relation predicting results P^{τ} for each relation type τ, as follows:

$$P^{\tau} = \gamma_m P_m^{\tau} + \gamma_y P_y^{\tau} + \gamma_a P_a^{\tau}, \tag{21}$$

where γ_m, γ_y and γ_a are weight coefficients to balance the importance of the DGPL and MGAT predictions, and $\gamma_m + \gamma_y + \gamma_a = 1$.

4.4 Training Strategy

Considering the imbalance of positive and negative samples in the dataset, we compute a weighted cross entropy loss $\mathcal{L}^{\tau}$ for each type of event relation as follows:

$$\mathcal{L}^{\tau} = -\frac{1}{K}\sum_{k=1}^{K}\left(\frac{n^{\tau}}{n_k^{\tau}}\right)^{\gamma} y_k^{\tau} log(\hat{y}_k^{\tau}) + \lambda||\theta||^2, \tag{22}$$

where y_k^{τ} and $\hat{y}_k^{\tau}$ are the gold label and the predicted relation of the k-th instance, n_k^{τ} and n^{τ} are the number of instances with the label y_k^{τ} and all instances for relation type τ, respectively. γ is the balance coefficient, K the batch size, λ and θ are the regularization hyper-parameters. The loss function of our PLAF is optimized as follows:

$$\mathcal{L} = \sum_{\tau\in\mathcal{S}} \mathcal{L}^{\tau}. \tag{23}$$

We use AdamW optimizer (Loshchilov and Hutter, 2017) with L2 regularization for model training.

5 Experiment

5.1 Experiment Settings

We conduct experiments on the constructed EGC-MAVEN dataset. Following the data splitting of MAVEN dataset (Tao et al., 2023), we split the EGC-MAVEN dataset into a training set (10,991 instances), a development set (3,589 instances), and a testing set (3,601 instances). We adopt the Precision (P), Recall (R) and F1-score (F1) as the evaluation metrics. More details about experimental settings can be found in Appendix C.

5.2 Competitors

We replicate some advanced knowledge graph completion models to conduct EGC as benchmarks, including HINGE (Rosso et al., 2020), ConGLR (Lin et al., 2022), APST (Su et al., 2024) and SIEG (Shi et al., 2024). Besides, we validate the effectiveness of large language models (LLMs) on the EGC task, including ChatGLM3-6B (Zeng et al., 2024), Llama3-7B (Dubey et al., 2024) and GPT-3.5-turbo (Ouyang et al., 2022). Details about the implementation of competitors and LLMs are in Appendix D and Appendix E, respectively.

5.3 Overall Results

Table 2 compares the overall EGC performance between our PLAF and the competitors on the EGC-MAVEN dataset. It can be observed that

Model	Sub-event			Causality			Temporal			Coreference			Average		
	R	P	F1	R	P	F1	R	P	F1	R	P	F1	R	P	F1
ChatGLM3-6B (Zeng et al., 2024)	18.0	8.7	10.3	47.6	13.4	20.9	24.6	7.1	11.1	46.2	25.3	32.7	34.1	13.6	18.7
Llama3-7B (Dubey et al., 2024)	47.6	11.7	14.4	46.6	13.9	19.8	41.1	16.1	21.4	91.3	15.8	27.0	56.7	14.4	20.6
GPT-3.5-turbo (Ouyang et al., 2022)	38.7	12.0	18.2	47.6	14.6	22.3	41.0	18.4	24.7	89.4	20.5	33.4	54.2	16.4	24.6
HINGE (Rosso et al., 2020)	80.7	15.4	25.6	69.1	16.6	26.4	55.3	16.0	24.0	89.6	33.3	48.5	73.7	20.3	31.1
ConGLR (Lin et al., 2022)	63.5	36.3	45.7	52.7	34.3	41.6	47.4	25.0	32.7	82.7	74.5	78.4	61.6	42.5	49.6
SIEG (Shi et al., 2024)	33.7	19.2	24.2	45.8	14.7	22.1	36.9	17.7	23.8	88.0	53.0	66.1	51.1	26.1	34.1
APST (Su et al., 2024)	67.9	22.1	32.8	72.8	25.8	38.0	55.9	23.7	32.9	95.1	59.0	72.8	72.9	32.6	44.1
PLAF	60.2	55.7	57.9	52.2	50.4	51.2	44.4	29.9	34.3	97.9	77.8	86.7	63.7	53.5	57.5

Table 2: Overall results of comparison models on the EGC-MAVEN dataset.

our PLAF has achieved significant performance improvements over all competitors, in terms of much higher F1 score on the completion of each relation type and the average results. We attribute the outstanding performance of our PLAF to: (1) Employing a Dual graph prompt learning to encode both semantic and structure information of EGs, based on event mention instance and event type schema respectively. It then uses an MLM to predict relation answer word(s) for missing relation identification; (2) Utilizing a multiplex graph attention network to facilitate learning the representations of each relation type through the interaction of other three relation types.

Besides, we can observe that none of LLM competitors (i.e. ChatGLM3, LLama3, GPT3.5) applying zero-shot learning can outperform the other well-trained competitors. This indicates that EGC is still a challenging task for the LLMs considering its complex input of both graph structure and event semantics. Further, we can observe that the competitors HINGE and SIEG cannot outperform ConGLR and APST, and the performance gaps are obvious. As HINGE only leverages the pairwise events' information and SIEG only utilizes some shortest path distance or number features, but neglecting the graph structure information of event graphs; While, ConGLR and APST has use the contextual graphs containing path nodes or relevant relation paths extracted from graphs, respectively, to focus on relation paths within the graph.

5.4 Ablation Study

Module Ablation: To examine the effectiveness of different modules, we design the following ablation study. PLAF *w/o* {*EvtMen., EvtType, MGAT*} is the full PLAF model without the predictions from either the event mention sequence or the event type sequence in DGPL module, as well as the prediction from the MGAT module, respectively.

Table 3 presents the results of our module ablation study. We can first observe that none of the PLAF *w/o EvtMen.*, PLAF *w/o EvtType*, and PLAF *w/o MGAT* can outperform the full PLAF model. This, however, is not unexpected. As the PLAF *w/o EvtMen.* ignores the semantic information of EGs, and the PLAF *w/o EvtType* ignores the schema information; While the PLAF *w/o MGAT* fails to learn relation representations interactively among different relation types. This also validates our argument that the identification of one type of missing relations may be facilitated by the information interaction from other types of event relations.

Relation Ablation: To further explore the facilitation of different relation types in missing relation prediction, we design the following relation ablation experiments. PLAF *w/o* τ is the PLAF model trained without relations of type τ; While PLAF *w/* τ is the PLAF model trained only using the relations of type τ, where $\tau \in \mathcal{S}$ and $\mathcal{S} = \{$Temp., Caus., SubE., Corf.$\}$.

Table 3 presents the results of our relation ablation study. We can observe that all PLAF *w/o* τ ablation models suffer from the absence of relation type τ. This indicates that each type of event relations in an KG are all critical, not only for the missing relation prediction of absence relation type, but also for the missing relation prediction of other types of relations. Besides, we can observe that the average performance decline of PLAF *w/o* Caus. and PLAF *w/o* Temp. is much more obvious than that of PLAF *w/o* SubE. and PLAF *w/o* Corf.. This might be attributed to the fact that causality relations contain the logical connection between events and temporal relations order the sequence of event occurrence, and they can provide crucial clues for the missing relation prediction.

Further, it is not unexpected that the performance

Model		Sub-event			Causality			Temporal			Coreference			Average		
		R	P	F1	R	P	F1	R	P	F1	R	P	F1	R	P	F1
PLAF *w/o*	MGAT	82.9	42.7	56.2	73.7	34.8	46.3	59.0	26.8	36.4	99.3	64.9	78.5	78.7	42.3	54.3
	EvtType	71.1	48.1	57.3	55.1	41.4	47.3	50.7	26.4	32.6	52.6	68.1	80.5	68.9	46.0	54.5
	EvtMen.	60.9	37.8	46.1	50.0	42.9	46.1	48.1	24.3	32.3	99.1	64.2	77.9	64.5	42.3	50.6
PLAF *w/o*	SubE.	43.7	36.0	38.9	52.3	44.5	48.0	39.4	28.1	32.7	97.7	77.8	86.6	58.2	46.6	51.6
	Caus.	49.4	41.8	45.2	43.6	30.8	36.0	32.4	25.2	28.3	98.8	75.2	85.4	56.0	43.2	48.7
	Temp.	44.4	38.8	39.9	40.9	41.5	41.0	26.6	22.6	24.4	95.3	71.4	81.6	51.8	43.5	46.7
	Corf.	57.1	56.7	56.9	55.5	45.5	50.0	36.9	29.8	32.9	98.8	73.3	84.2	62.1	51.3	56.0
PLAF *w/*	SubE.	63.7	34.7	44.6	47.4	11.4	18.2	41.4	17.9	13.9	10.3	2.0	10.3	40.7	16.5	21.8
	Caus.	58.2	8.2	14.1	52.6	30.0	38.1	35.2	20.5	13.4	51.4	26.4	34.8	49.4	21.3	25.1
	Temp.	46.9	9.0	11.7	37.9	9.2	13.7	19.7	27.9	23.1	8.0	2.2	3.5	28.1	12.1	13.0
	Corf.	39.1	6.0	9.4	49.9	12.5	19.5	19.3	11.4	14.3	92.0	82.4	86.9	50.1	28.1	32.5
PLAF		**60.2**	**55.7**	**57.9**	**52.2**	**50.4**	**51.2**	**44.4**	**29.9**	**34.3**	**97.9**	**77.8**	**86.7**	**63.7**	**53.5**	**57.5**

Table 3: Experiment results of module ablation study on the EGC-MAVEN dataset.

of our PLAF has declined very severely when only using one type of relations for model training. As they can not capture information from other unseen relation types, so it only performs well in predicting the trained relation type, but struggles to make accurate predictions of other relation types. Nevertheless, we can observe that the performance of PLAF *w/* τ, when predicting the trained relation type τ, is comparable to PLAF *w/o* τ which is trained with other three relation types but without the predicting one. (As colored in Table 3). This again validated our arguments that identifying one type of missing relations may be benefited from the other three types of existing relations.

Coefficient Ablation: Fig. 4 plots the performance of our PLAF against using different balance coefficient of cross-entropy loss in Formula 22, where the larger the coefficient, the greater the difference in loss between positive and negative samples. We can observe that our PLAF achieves the best performance in terms of the F1 score when the balance coefficient γ is set to 0.8, and it suffers from either a very large or very small value.

On the one hand, a very small balance coefficient γ ignores the imbalance issue between positive and negative samples, which can be significantly improved when it is set to a larger value. This validates the effectiveness of the weighted cross-entropy loss we applied. On the other hand, a very large balance coefficient γ will impose stricter penalties for mispredictions of positive samples, so as to relaxes the requirement for completion predictions, which thus admits too many assumed missing relations to be accepted. As such, the recall is higher yet the precision is lower, resulting the F1 score slightly decreases.

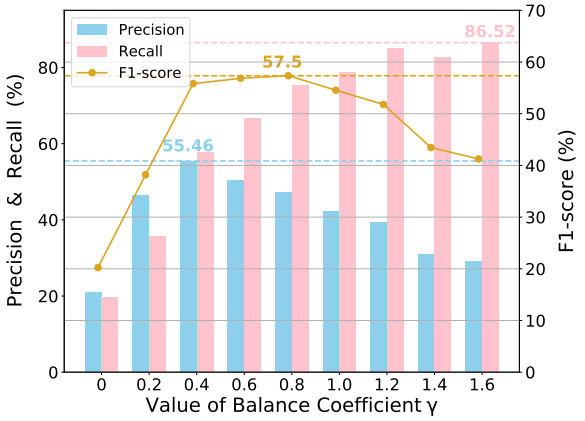

Figure 4: Performance comparison of coefficient ablation study on the EGC-MAVEN dataset.

6 Concluding Remarks

This paper introduces an event graph completion task that aims to complete multi-relation edges between event nodes in heterogeneous EGs. We propose a Multiplex Graph Prompt Learning and Attentive Fusion (PLAF) model, which employs a dual graph prompt learning to encode both semantic and structural information of EGs and utilizes a multiplex graph attention network to facilitate the interaction among different relation types. Finally, the predictions from DGPL and MGAT modules are aggregated to complete the EGs. We conduct experiments on our constructed EGC-MAVEN dataset and experimental results have validated the effectiveness of our proposed method.

7 Limitations

• Due to the lack of annotated event arguments in the MAVEN-ERE dataset, this paper solely focuses on event information, neglecting the influence of event arguments. The absence of these arguments may lead to a decline in model performance for event predictions.

• During the linearization process, due to the limit of PLM length, some detailed information may be lost, which affects the model's understanding and predictive ability regarding events, potentially leading to the neglect of important information.

References

Snigdha Chaturvedi, Haoruo Peng, and Dan Roth. 2017. Story comprehension for predicting what happens next. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1603–1614.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Paraskevas Dimitriou and Vasileios Karyotis. 2024. Empowering random walk link prediction algorithms in complex networks by adapted structural information. *IEEE Access*.

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2021. Prompt-learning for fine-grained entity typing. *arXiv preprint arXiv:2108.10604*.

Chenyang Du, Xiaoge Li, and Zhongyang Li. 2024. Semantic-enhanced reasoning question answering over temporal knowledge graphs. *Journal of Intelligent Information Systems*, pages 1–23.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Goran Glavaš and Jan Šnajder. 2015. Construction and evaluation of event graphs. *Natural Language Engineering*, 21(4):607–652.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Muhammed Ifte Khairul Islam, Khaled Mohammed Saifuddin, Tanvir Hossain, and Esra Akbas. 2024. Dygcl: Dynamic graph contrastive learning for event prediction. *arXiv preprint arXiv:2404.15612*.

Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Sujian Li, Baobao Chang, and Zhifang Sui. 2016. Encoding temporal information for time-aware link prediction. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 2350–2354.

Xiaomeng Jin, Manling Li, and Heng Ji. 2022. Event schema induction with double graph autoencoders. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2013–2025.

Manling Li, Sha Li, Zhenhailong Wang, Lifu Huang, Kyunghyun Cho, Heng Ji, Jiawei Han, and Clare Voss. 2021a. The future is not one-dimensional: Complex event schema induction by graph modeling for event prediction. *arXiv preprint arXiv:2104.06344*.

Manling Li, Tengfei Ma, Mo Yu, Lingfei Wu, Tian Gao, Heng Ji, and Kathleen McKeown. 2021b. Timeline summarization based on event graph compression via time-aware optimal transport. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6443–6456.

Qing Li, Tao You, Jinchao Chen, Ying Zhang, and Chenglie Du. 2024. Biodyngrap: Biomedical event prediction via interpretable learning framework for heterogeneous dynamic graphs. *Expert Systems with Applications*, 244:122964.

Qika Lin, Jun Liu, Fangzhi Xu, Yudai Pan, Yifan Zhu, Lingling Zhang, and Tianzhe Zhao. 2022. Incorporating context graph with logical reasoning for inductive relation prediction. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 893–903.

Jiahao Liu, Dongsheng Li, Hansu Gu, Tun Lu, Peng Zhang, and Ning Gu. 2022. Parameter-free dynamic graph embedding for link prediction. *Advances in Neural Information Processing Systems*, 35:27623–27635.

Lihui Liu, Blaine Hill, Boxin Du, Fei Wang, and Hanghang Tong. 2024. Conversational question answering with language models generated reformulations over knowledge graph. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 839–850.

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. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Xin Ren, Luyi Bai, Qianwen Xiao, and Xiangxi Meng. 2023. Hierarchical self-attention embedding for temporal knowledge graph completion. In *Proceedings of the ACM Web Conference 2023*, pages 2539–2547.

Paolo Rosso, Dingqi Yang, and Philippe Cudré-Mauroux. 2020. Beyond triplets: hyper-relational knowledge graph embedding for link prediction. In *Proceedings of the web conference 2020*, pages 1885–1896.

Bin Shang, Yinliang Zhao, and Jun Liu. 2024. Learnable convolutional attention network for knowledge graph completion. *Knowledge-Based Systems*, 285:111360.

Tong Shen, Fu Zhang, and Jingwei Cheng. 2022. A comprehensive overview of knowledge graph completion. *Knowledge-Based Systems*, 255:109597.

Lei Shi, Bin Hu, Deng Zhao, Jianshan He, Zhiqiang Zhang, and Jun Zhou. 2024. Structural information enhanced graph representation for link prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 14964–14972.

Zhixiang Su, Di Wang, Chunyan Miao, and Lizhen Cui. 2024. Anchoring path for inductive relation prediction in knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9011–9018.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Zhengwei Tao, Zhi Jin, Xiaoying Bai, Haiyan Zhao, Chengfeng Dou, Yongqiang Zhao, Fang Wang, and Chongyang Tao. 2023. Seag: Structure-aware event causality generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4631–4644.

Takanori Ugai, Shusaku Egami, and Ken Fukuda. 2024. Event prediction in event-centric knowledge graphs using bert. In *2024 IEEE 18th International Conference on Semantic Computing (ICSC)*, pages 306–310. IEEE.

Chengyu Wang, Jianing Wang, Minghui Qiu, Jun Huang, and Ming Gao. 2021. Transprompt: Towards an automatic transferable prompting framework for few-shot text classification. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 2792–2802.

Hongwei Wang, Zixuan Zhang, Sha Li, Jiawei Han, Yizhou Sun, Hanghang Tong, Joseph P Olive, and Heng Ji. 2022a. Schema-guided event graph completion. *arXiv preprint arXiv:2206.02921*.

Jingchao Wang, Weimin Li, Wei Liu, Can Wang, and Qun Jin. 2023. Enabling inductive knowledge graph completion via structure-aware attention network. *Applied Intelligence*, 53(21):25003–25027.

Xiaozhi Wang, Yulin Chen, Ning Ding, Hao Peng, Zimu Wang, Yankai Lin, Xu Han, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022b. MAVEN-ERE: A unified large-scale dataset for event coreference, temporal, causal, and subevent relation extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 926–941, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiyao Wei, Saiping Guan, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2024. Few-shot link prediction on hyper-relational facts. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7196–7207.

Zhankun Xiong, Shichao Liu, Feng Huang, Ziyan Wang, Xuan Liu, Zhongfei Zhang, and Wen Zhang. 2023. Multi-relational contrastive learning graph neural network for drug-drug interaction event prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5339–5347.

Weijian Yu, Jie Yang, and Dingqi Yang. 2024. Robust link prediction over noisy hyper-relational knowledge graphs. In *The Web Conference 2024*.

Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Xun Zhu, Wang Gao, Tianyu Li, Wenguang Yao, and Hongtao Deng. 2024. Event-centric hierarchical hyperbolic graph for multi-hop question answering over knowledge graphs. *Engineering Applications of Artificial Intelligence*, 133:107971.

A Relation Optimize

The MAVEN-ERE dataset annotated relations at the event pair level, without considering the global structure of entire event graphs. Therefore, as illustrated in Fig. 2(b), we optimized the constructed graph in two aspects: Coreference Replicate and Temporal simplify.

Coreference Derivation: Considering that events with coreference relations should have the same temporal, causality, and sub-event relations, we further derive more event relations based on the coreference relation in each EG. As shown in the top part of Fig. 2(b), if the *subdue* event has a causality relation with the *brought* event, its coreference event *conquer* should also has a causality relation with the *brought* event. Then we add a new relation ($brought \overset{Caus.}{\rightarrow} conquer$) to the constructed EGs.

Algorithm 1 Temporal Simplify Algorithm

Input: $\mathcal{R}^{\text{Temp.}}$: initial EG, including all temporal relations in $\mathcal{R}$ annotated by MAVEN-ERE.
Output: $\hat{\mathcal{R}}$: simplified temporal relations of the EG, some temporal relations are removed.

```
Initial R̂ = R^Temp
for r_ij in R^Temp do
    R̂ = R̂ / r_ij
    if r_ij can NOT be inferred by R̂ then
        R̂ = R̂ + r_ij
    end if
end for
return R̂
```

Temporal Simplify: MAVEN-ERE annotates temporal relations for all event pairs, resulting in a complex networked event temporal structure. We argue that representing event timelines in a chain-like manner offers clearer representation. Therefore, we simplify the intricate networked temporal relations into a set of chain-like temporal relations. As shown in the middle part of Fig. 2(b), the temporal relation *plebiscite* $\rightarrow$ *brought* is removed because it can be inferred by the temporal relation *plebiscite* $\rightarrow$ *concluded* $\rightarrow$ *brought*.

As illustrated in Algorithm 1, we check whether each temporal relation r_{ij} in $\mathcal{R}^{\text{Temp.}}$ can be inferred from other relations within $\mathcal{R}^{\text{Temp.}}$. If so, we delete that relation r_{ij}; otherwise, we retain it. This process yields a new relation set $\hat{\mathcal{R}}$, where some redundant temporal relations have been eliminated. Through this algorithm, we optimize the intricate network of temporal relations into a combination of multiple chain-like temporal relations, better reflecting the logical connections and developmental patterns of events.

B Example of Linearization

To enhance the clarity of how our Dual Graph Prompt Learning module transforms the graph into event mention and event type templates, we offer a comprehensive example depicted in Fig. 5. The upper part of Fig. 5 showcases an instance selected from our EGC-MAVEN dataset, including an event graph and a query. The middle part of Fig. 5 illustrates all the event triplets from the input EG. And the bottom part of Fig. 5 illustrates event mention template and event type template.

Initially, we transform the input EG into event triplets, denoted as $T_k = (e_m, r^{\tau}_{mn}, e_n)$, and employ our proposed Distance-First Linearization (DFL) approach to order these event triplets according to their distance to the query triplet $T_q = (e_i, r^{\tau}_{ij}, e_j)$. When ordering triplets for event mention template, the distance of an event triplet T_k to the query triplet T_q is determined by the number of edges on the shortest undirectional path from either one of its event nodes $e_a \in \{e_m, e_n\}$ to either one of the query's event nodes $e_b \in \{e_i, e_j\}$ as shown in Formulation 2. For event type template, we employ a similar distance calculation method, with the exception that the distance between two events of identical event types is set to zero, i.e. $\min_{p \in P(murder, executed)} |p| = 0$. Building on this, we argue the closer an event triplet T_k is to the query triplet T_q, the more critical information it can provide for its relation completion. Consequently, we order these event triplets in descending order of their distances to the query triplet to construct the event mention and event type templates, as shown in the bottom of Fig. 5.

C Experiment Settings

Our method is implemented based on the pre-trained RoBERTa-base model (Liu et al., 2019) with 768-dimension provided by HuggingFace transformers [2], and run PyTorch framework with CUDA on NVIDIA GTX 3090 GPU. We use RoBERTa as our Sequence Encoder and Context Encoder. We set the learning rate for the PLM to 5e-6. Intra-coefficient α and cross-coefficient β are set to 1 and 0.5, respectively. The batch size is set to 1.

[2] https://github.com/huggingface/transformers

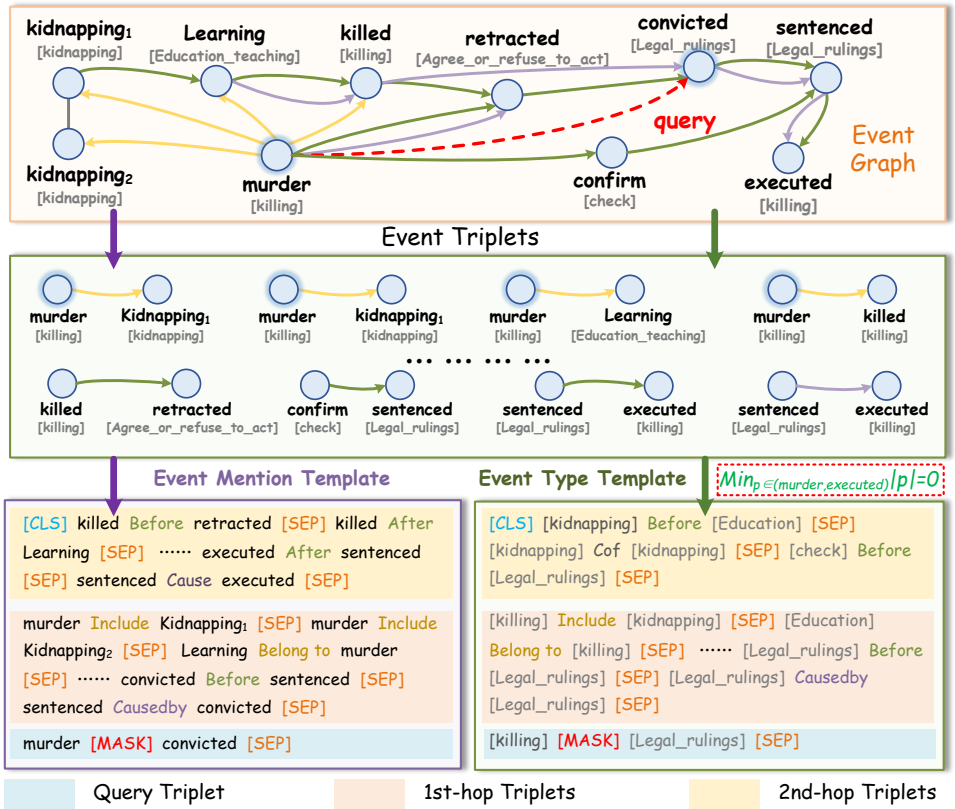

Figure 5: A detailed example for the Linearization process of Dual Graph Prompt Learning module.

The MLP is a two-layer fully connected network, in which the activation function is LeakyReLU and the rate of dropout is 0.4. And when calculating the loss $\mathcal{L}^{\tau}$ of each type τ, the balance coefficient γ is set to 1.5.

We evaluate the model performance using Precision (P), Recall (R), and macro-F1 (F1). Specifically, we calculate the macro-averaged results for Precision, Recall, and F1-score for the relations of different directions. Additionally, we use the average P, R, and F1 of different type relations as the performance evaluation metrics for the entire dataset, represent as follows:

$$P = \frac{1}{N} \sum\nolimits_{\tau \in \mathcal{C}} P^{\tau}, \tag{24}$$

$$R = \frac{1}{N} \sum\nolimits_{\tau \in \mathcal{C}} R^{\tau}, \tag{25}$$

$$F1 = \frac{1}{N} \sum\nolimits_{\tau \in \mathcal{C}} F1^{\tau}, \tag{26}$$

where P^{τ}, R^{τ} and $F1^{\tau}$ are the evaluation metrics for relations of each type τ and N is the number of types.

D Competitors

We replicate four advanced knowledge graph completion models as benchmarks, including HINGE (Rosso et al., 2020), ConGLR (Lin et al., 2022), APST (Su et al., 2024) and SIEG (Shi et al., 2024). The introductions of these competitors are as follows:

• HINGE (Rosso et al., 2020) proposes a hyper-relational knowledge graph embedding model to capture both structural information of KGs encoded in triplets and the correlation between triplets. Specifically, it addresses the complexity of hyper-relational data that is often overlooked by existing knowledge graph embedding techniques, which typically represent knowledge graphs as sets of triplets. Hyper-relational data includes not only the base triplet (h, r, t) but also associated key-value pairs (k, v). HINGE learns directly from these hyper-relational facts, capturing both the primary structural information encoded in the triplets and the correlation between each triplet and its associated key-value pairs.

• ConGLR (Lin et al., 2022) employs two GCNs with the information interaction to process the extracted enclosing graph and introduced context graph, and integrates neural calculating and logical reasoning to obtain inductive scores. Specifically, it addresses integrating a context graph to acquire relation embeddings and represent logical rules. The ConGLR firstly extract the enclosing subgraph of

Input for LLMs

Description:

We annotated the events and their relationships in the given document , and constructed an event graph with the events as nodes and the relations between them as edges . There are four types of relations: Sub-event, Temporal, Causal, Coreference.

Sentence:

- **Expedition:** The Expedition of the Thousand was an event of the Italian Risorgimento that took place in 1860.
- **sailed:** A corps of volunteers sailed from Quarto.

.....

- **brought:** The expedition was a success and concluded with a plebiscite that brought Naples and Sicily into the Kingdom of Sardinia.

Structure:

Sailed Before landed [SEP] landed Before conquer [SEP] ····· Expedition Cof expedition [SEP] Expedition Before plebiscite [SEP] Expedition Include sailed [SEP]

Question:

Please determine if there is a missing Temporal relation between Event1 Expedition and Event2 brought. If there is, respond with Before or After (Before means Event1 occurs before Event2, while After means Event2 occurs before Event1 in time) . If there is no such relationship, respond with None. Please respond with a single word or phrase .

Figure 6: Illustration of LLMs input.

target nodes and initialize their representations by the double radius labeling. Secondly, it introduces the context graph including three types of nodes (relational path, relation and entity) to meet the requirements of relation modeling and logical reasoning. Finally, two GCNs are carried out to process the subgraph and context graph respectively to generate reasoning vectors for logical score calculation.

• SIEG (Shi et al., 2024) proposes a structural information enhanced link prediction framework to fit neighborhood graph structures and introduce a Binary Structural Transformer (BST) to encode the structural relationships between target nodes. Specifically, it addresses two key limitations of Graph Neural Networks (GNNs) in link prediction tasks: the inability to perceive long-range structural information due to restricted receptive fields and the potential information loss when focusing on node-centric rather than link-centric tasks. On one hand, we use a GNN to collect the neighborhood information of target nodes, removing neighbor node features to make the model more focused on structural features. On the other hand, the BSA is introduced to encode the pairwise features of the target nodes, especially the long-range structural features, to complement the deficiency of GNN.

• APST (Su et al., 2024) proposes an assessment mechanism to evaluate the rationality of paths, and utilizing logically irrelevant ones to enhance predictions. Specifically, the paper points out that traditional path-based methods rely heavily on CPs, which can limit their performance in incomplete KGs where CPs between entities are scarce. To solve the limitation, it enhances the prediction process by introducing Anchoring Paths (APs), which are paths connecting to either the head or tail entity of a query triplet. Moreover, APST retrieves detailed descriptions from public websites to enrich the semantic meaning of entities and relations, and employs AP accuracy and AP recall as metrics to filter out irrelevant APs, ensuring that only logical paths are considered for prediction.

E Input for LLMs

When exploring the performance of LLMs on our proposed EGC task, we developed new prompt templates to convert event graphs into text sequences as input into LLMs. As illustrated in Fig. 6, we concatenate the description of event graphs, the contextual sentences for each event node, the linearized event graph structure, and the question to be asked as input for the LLM. For each instance, we ask the LLMs four times to determine whether there is any missing relations of type τ between the query nodes. Due to page limits, we only show the input for determining whether there is a missing temporal relation between query events in the Fig. 6.

F Few shot

Some researchers have reported the robustness of prompt learning paradigm in using fewer training data (Wang et al., 2021; Ding et al., 2021). Since our PLAF model also employs the prompt learning paradigm to make prediction, we examine its performance in low-resource scenarios, as shown in the Fig. 7.

As expected, the performance of PLAF model gradually decreases as the amount of training data decreases. However, the decrease in performance is very slow, with an F1 score decrease of about 13% when training data is reduced by 60%. Notably, even with only 20% of the training data, PLAF (F1: 42.6%) outperforms many competitors with full training data such as SIEG (34.1%). These results confirm the effectiveness of PLAF even with fewer training data, benefiting from the abundant knowledge embedded in the advanced PLMs. Additionally, we also observe that the model performs poorly in zero-shot scenarios. This aligns with our

Model	Sub-event			Causality			Temporal			Coreference			Average		
	R	P	F1	R	P	F1	R	P	F1	R	P	F1	R	P	F1
PLAF *BERT*	50.7	59.9	54.5	46.2	44.6	45.3	39.5	25.6	30.3	95.5	76.5	85.0	58.0	51.6	53.8
PLAF *ERNIE*	84.7	34.4	48.2	74.1	37.7	50.0	58.6	28.4	38.2	96.7	75.3	84.7	78.5	44.0	55.3
PLAF *DeBERTa*	57.4	66.3	61.0	58.7	45.5	51.2	39.0	28.4	32.9	99.1	69.2	81.5	63.5	52.3	56.6
PLAF *RoBERTa*	60.2	55.7	57.9	52.2	50.4	51.2	44.4	29.9	34.3	97.9	77.8	86.7	63.7	53.5	57.5

Table 4: PLM Ablation results on the EGC-MAVEN dataset.

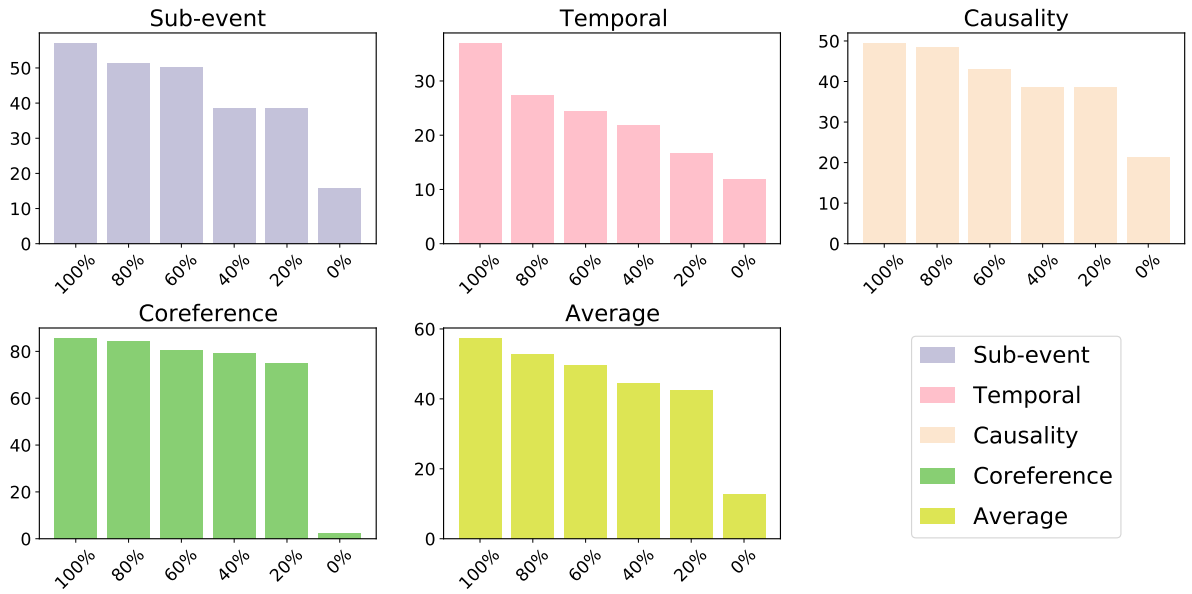

Figure 7: Results of our PLAF model on few-shot Scenarios.

observation in Section 5.3 that even larger language models in zero-shot scenarios exhibit a significant performance decline in event graph relation completion compared to other well-trained competitors. That's because, without training, it is still challenging for prompt-based methods to directly comprehend the event sequences transformed from graphs and complete missing relations.

G PLM study

Our PLAF model is sensitive to PLM. To explore the impact of using different PLMs on model performance and select the most suitable one, we conducted an ablation experiment with several PLMs, as shown in 4. The introductions of main PLMs we considered are as follows:

- **BERT** (Devlin et al., 2018): The most representative PLM proposed by Google[3], which is pre-trained using a cloze task and a next sentence prediction task.
- **RoBERTa** (Liu et al., 2019): A BERT enhanced PLM proposed by Facebook[4], which removes the next sentence prediction objective and is pre-trained on a much larger dataset with some modified key hyper-parameters.
- **ERNIE** (Sun et al., 2019): A knowledge enhanced PLM proposed by Baidu[5], which uses some knowledgeable masking strategies in pre-training.
- **DeBERTa** (He et al., 2020): The latest masked PLM proposed by Microsoft[6], which improves BERT and RoBERTa models using a disentangled attention mechanism and an enhanced mask decoder.

We can observe that our PLAF model with different PLMs has achieved significant performance improvements overall competitors. We attribute its outstanding performance to the graph prompt

[3] https://github.com/google-research/bert

[4] https://github.com/pytorch/fairseq/

[5] https://github.com/PaddlePaddle/ERNIE

[6] https://github.com/microsoft/DeBERTa

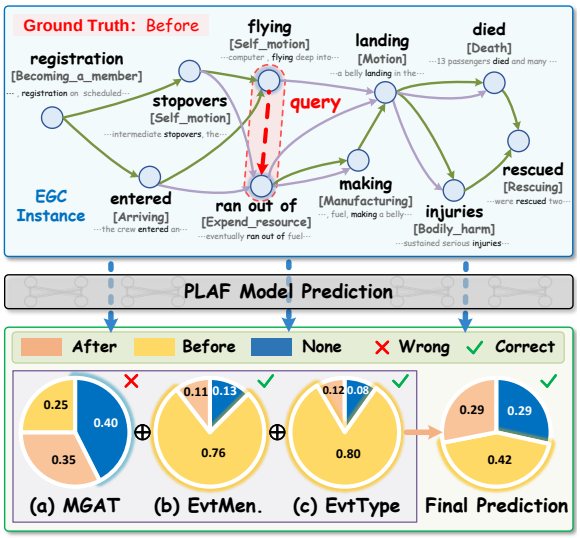

Figure 8: A case of prediction probabilities of each module in our PLAF model. The figure shows only the prediction results of temporal type relations.

learning, encoding both semantic and structural information of the EG, and the additional focus on the interaction among various types of events.

We can also observe that using different PLMs do result in some performance variations. This is not unexpected. It can be attributed to that while all the four PLMs employ a kind of Transformer-based model in pre-training on large-scale corpus, their training strategies or training corpus are not entirely identical. Compared to PLAF-*BERT*, our PLAF model using ERNIE, DeBERTa, or RoBERTa achieved better performance. This is attributed to the fact that these three PLMs have made some optimizations based on BERT. For example, ERNIE utilizes a strategy of continuous learning in the pre-training stage. Finally, PLAF-*RoBERTa* achieved the best performance, which removes the next sentence prediction objective and is pre-trained on a much larger dataset with some modified key hyper-parameters. Therefore, we implement the remaining ablation experiments with RoBERTa.

H Case Study

To visually demonstrate the effectiveness of our designed DGPL module, in Fig. 8, we present a case study illustrating the detailed prediction probabilities of each module for an EGC instance. The upper part of Fig 8 depicts an EGC instance where the ground truth for temporal relation completion is 'Before', indicating that the event 'flying' occurs before the event 'run out of'. The lower part of Fig. 8 shows the predictions for temporal relation completion of each module, including (a) MGAT P_a^{cau}, (b) *EvtMen.* P_m^{cau}, (c) *EvtType* P_y^{cau} and (d) final prediction P^{cau}.

In this instance, the MGAT module predicts an incorrect result ('None'), while both the *EvtMen.* and *EvtType* modules predict the correct result ('Before'). Following the corrections made by the *EvtMen.* and *EvtType*, the model's final prediction was adjusted to the correct result ('Before'). If there is no temporal relation between events 'flying' and event 'run out of', the event graph will be divided into two temporally unrelated subgraphs. However, since there are other relations, such as causal relations, between the events in the two subgraphs, it is easy to infer a temporal relation between the two events based on event chain-level information. Compared to the MGAT module based on a graph attention mechanism, focusing mainly on neighbor information, our *EvtMen.* and *EvtType* modules is better at modeling event path information. It is the path information enables to capture the interactions between two subgraphs and infer the temporal relation between the query nodes.