Semantic and Sentiment Dual-Enhanced Generative Model for Script Event Prediction

Feiyang Wu[1], Peixin Huang[2], Yanli Hu[2] , Zhen Tan[2], Xiang Zhao[1]*
[1]Laboratory for Big Data and Decision,
National University of Defense Technology, China
[2]National Key Laboratory of Information Systems Engineering,
National University of Defense Technology, China
{wufeiyang,huangpeixin15,huyanli,tanzhen08a,xiangzhao}@nudt.edu.cn

Abstract

Script Event Prediction (SEP) aims to forecast the next event in a sequence from a list of candidates. Traditional methods often use pre-trained language models to model event associations but struggle with semantic ambiguity and embedding bias. Semantic ambiguity arises from the multiple meanings of identical words and insufficient consideration of event arguments, while embedding bias results from assigning similar word embeddings to event pairs with similar lexical features, despite their different meanings. To address above issues, we propose a the Semantic and Sentiment Dual-enhanced Generative Model (SSD-GM). SSD-GM leverages two types of script event information to enhance the generative model. Specifically, it employs a GNN-based semantic structure aggregator to integrate the event-centric structure information, thereby mitigating the impact of semantic ambiguity. Furthermore, we find that local sentiment variability effectively reduces biases in event embeddings, while maintaining global sentiment consistency enhances predictive accuracy. As a result, SSD-GM adeptly captures both global and local sentiment of events through its sentiment information awareness mechanism. Extensive experiments on the Multi-Choice Narrative Cloze (MCNC) task demonstrate that our approach achieves better results than other state-of-the-art baselines.

1 Introduction

Script represents a form of structured knowledge derived from textual data, encompassing a sequence of events. Recent attention has focused on narrative event chains(Chambers and Jurafsky, 2008), where events involving a common participant (the protagonist) are ordered temporally. Script Event Prediction (SEP) uses this structure to predict the next event in a sequence from a list of candidates. Figure 1 presents a restaurant script involving sequences of events that occurs when a person orders the food. This approach supports various NLP tasks, such as discourse understanding(Lee and Goldwasser, 2019) and story generation(Chaturvedi et al., 2017). Early works (Granroth-Wilding and Clark, 2016; Wang et al., 2018) used embedding methods (e.g., Word2Vec) to calculate similarities between event candidates and script events. After that, some works(Lv et al., 2020) attempted to leverage masked language model (MLM) (Liu, 2019) to obtain better event representations and they all achieved a substantial improvement.

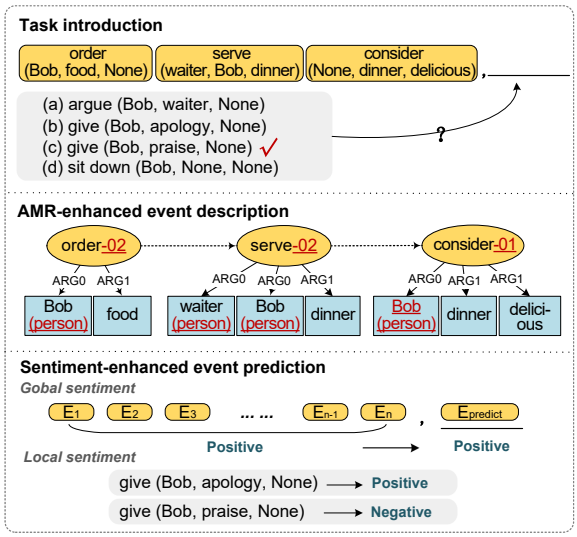

Figure 1: An example of a simplified SEP task that describes how semantic structure and sentiment information enhance the SEP task. AMR enriches the representation of events by parsing out additional semantic information marked in red. The global sentiment consistency contributes to event prediction, and local sentiment can address event embedding biases.

Despite promising results achieved by these methods, some challenges still remain, particularly concerning event semantic ambiguity and event embedding bias. While pre-trained models offer some disambiguation, they are not fully effective alone. Through further research, we found that incorporat-

*Corresponding author.

Proceedings of the 31st International Conference on Computational Linguistics, pages 9250–9259
January 19–24, 2025. ©2025 Association for Computational Linguistics

ing semantic structure and sentiment information significantly improves handling these issues.

Regarding event semantic ambiguity, existing methods represent each event with a verb and three arguments(e.g., subject, object, and indirect object). But it often fails to capture the full meaning of events due to three main limitations:(1) Ambiguity in verbs can lead to misunderstandings. For instance, in Figure 1, the verb "order" can be interpreted as either a "sequence" or a "command," making the intended meaning unclear. (2) Headwords do not always adequately describe the participant types. Although some methods might infer participant types from headwords, they often fail to distinguish between entities like "Bob" and "waiter" in the event "Waiter served Bob dinner." (3) Missing arguments can obscure event meaning; for instance, "felt dinner delicious" lacks the subject "Bob", leading to incorrect inferences by models. Abstract Meaning Representation (AMR) (Banarescu et al., 2013) parsing addresses these issues by offering clearer semantic distinctions and identifying missing elements. For example, AMR parses "order" as "01-ordering food" and identifies "Bob" and "waiter" correctly, also recovering missing subjects like "Bob."

Event embedding bias arises because pre-trained models often reflect only superficial semantics. As a result, event pairs with high lexical overlap may have similar embeddings despite differing meanings. Incorporating sentiment helps differentiate between events. For instance, "Bob give an apology" and "Bob give a praise" have different sentiments, and incorporating sentiment into event representations helps differentiate between these events.

In this paper, we propose the Semantic and Sentiment Dual-enhanced Generative Model (SSD-GM). To tackle semantic ambiguity, SSD-GM employs AMR to provide clearer event representations and correct missing elements. We utilize AMR tools to convert event sequences into AMR graphs. Then, we employ a Relational Graph Convolutional Network (RGCN)-based semantic relation aggregation module to consolidate semantic structure information. This semantic relation aggregation module maintains consistency between semantic structures and basic sentences while integrating global features by propagating local information along the graph paths. We incorporate a heterogeneous information fusion mechanism which optimizes information from diverse sources. To resolve embedding bias, SSD-GM incorporates a sentiment information awareness module that captures both global and local sentiments, mitigates embedding biases through local sentiment variance, and guides the prediction of subsequent events through sentiment consistency. To summarize, our main contributions are highlighted as follows:

• We propose SSD-GM, a novel model that combines semantic structure and sentiment awareness to tackle event semantic ambiguity and embedding bias in SEP.

• We introduce a semantic structure aggregator that combines original sentences with AMR for clearer semantics and reduced ambiguity. Our sentiment awareness module further captures global and local sentiments to improve differentiation and address embedding bias.

• Experimental results on the multi-choice narrative cloze (MCNC) task show that our approach outperforms existing state-of-the-art baselines.

2 The Proposed Approach

Script event prediction involves forecasting the most probable subsequent event given a sequence of events. Formally, with a script $X = \{x_1, \ldots, x_n\}$ and event candidates $Y = \{y_1, \ldots, y_m\}$, where x_i and y_j represent events, the objective is to select the most likely subsequent event y_t from Y. Each event x_i is defined by a tuple $x_i = (a_v^i, a_s^i, a_o^i, a_p^i)$, including four arguments: subject a_s, verb a_v, object a_o, and indirect object a_p (Granroth-Wilding and Clark, 2016). If any argument is missing, it is represented as *null*. For instance, the event tuple $x = \text{serve}(\text{waiter}, \text{bob}, \text{dinner})$ conveys "The waiter serves Bob dinner." In this task, the model is designed to learn and compute the conditional probability distribution $P(y_i \mid X)$ for each event candidate y_i $(i \in 1, \ldots, m)$, given the script X. Consequently, considering that generative pre-trained models have achieved remarkable advancements in nlp downstream tasks, we leverage Contrastive Fine-Tuning BART (Zhu et al., 2023) as the foundational architecture for our model to effectively model the conditional probability distribution $P(Y \mid X)$.

In this section, we introduce the proposed model. Figure 3 illustrates the overall architecture of the SSD-GM model.

2.1 Script Event Encoding

For a more comprehensive application of the script, we concatenate the script events to form an event sequence $S = \{x_1, x_2, \ldots, x_n\}$. We then use the BART encoder to obtain the embedding for each word in the event sequence. BART combines elements from both bidirectional and autoregressive

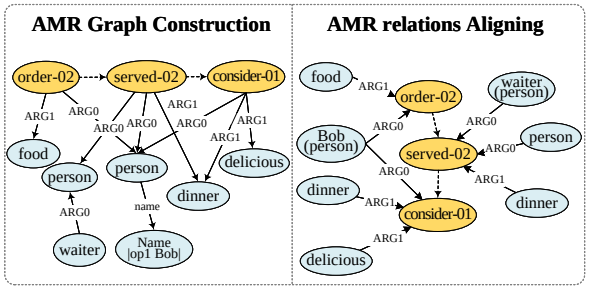

Figure 2: An example of AMR Preprocessing, where yellow represents the verb node and blue represents other event elements.

Semantic Roles	Types
ARG0, ARG1, ARG2, ...	Core Roles
op1, op2, op3, op4	Operators
manner, instrument, topic, ...	Means
time, year, weekday,...	Temporal
Other semantic roles	Others

Table 1: Semantic roles in the AMR graphs

models. By encoding information bidirectionally, BART can capture context from both directions in a sentence, while its autoregressive decoding allows for effective generation of coherent and contextually relevant text. The sequence S is initially formatted as "<s>x_1<SEP>x_2<SEP>...x_n<s>," with each event separated by the <SEP> token. Since the <SEP> token is not available in the BART model, we use "." as the separator. The output embedding (last hidden states) from the BART encoder is also the resulting event sequence embedding and is denoted as $H = \{h_1, h_2, ..., h_n\}$.

2.2 Semantic Structure Aggregator

The central tenet of the Semantic Structure Aggregator is to enhance the semantic enrichment of event embeddings through the incorporation of explicit semantic structural information, thereby effectively mitigating event semantic ambiguity.

2.2.1 AMR Preprocessing

Following this, we elaborate on the details of the Semantic Structure Aggregator, including AMR preprocessing, the semantic relation aggregate, and the heterogeneous information fusion.

AMR Graph Construction: To leverage the semantic structure of AMR for script event prediction, we first employ an AMR parser to convert the event sequence text into an AMR graph. As shown in Figure 2, it contains fine-grained node and edge types. We use the off-the-shelf parser SPRING (Bevilacqua et al., 2021) to ensure high-quality AMR outputs. In the AMR graph, the nodes indicate specific semantic elements and the edges indicate the semantic roles among them. Table 1 lists the semantic roles used in the AMR graph.

AMR Relations Aligning: We then add inverse edges to all the edges in the AMR graph to form the final semantic graph, making it reachable between any two nodes. Then, we align the parsed AMR using the aligner LEAMR(Blodgett and Schneider, 2021). This alignment allows us to reconstruct the AMR relations between words in event sequences, resulting in a transformed AMR where words serve as nodes.

Embedding: Once alignment is complete, we have transformed AMRs, which can be considered as event sequences annotated with AMR relations. We need to obtain their embedding to facilitate later representation learning by the model. For words in the event sequence, which are also the nodes in the AMR, we use BERT (Devlin, 2018) to encode the words (nodes) in the event sequence to obtain embeddings for subsequent representation learning. For the edges in the AMR, we represent the relationships between nodes using an adjacency matrix $R = \{r_{ij} \mid 1 \leq i, j \leq n\}$, where r_{ij} is the embedding of the edge label between words v_i and word v_j. Edge label embeddings are also obtained from the pre-trained model.

2.2.2 Semantic Relation Aggregate

SEP requires the model to fully understand what each event describes. To model the semantic parameters and associative relationships of events, SSDGM aggregates event-centric information from the constructed semantic structure graph.

The graph convolutional network excels at aggregating information from neighboring nodes, which suits the semantic structure of data. We use a RGCN as proposed by Schlichtkrull et al. (2018) to consolidate semantic information from surrounding nodes. The RGCN updates node representations with:

$$h_i^{l+1} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in N_i^r}\frac{1}{c_{i,r}}W_r^l h_j^l + W_0^l h_i^l\right)$$

Here, $\mathcal{R}$ is the set of relationship types, N_i^r denotes neighbors connected by relationship r, $c_{i,r}$ is the size of N_i^r, and W_r^l and W_0^l are trainable parameters. The function σ is a non-linear activation function. The initial feature vectors of the nodes are given by h_i^0 and h_j^0. This layer-wise propagation process updates each node's features based on local

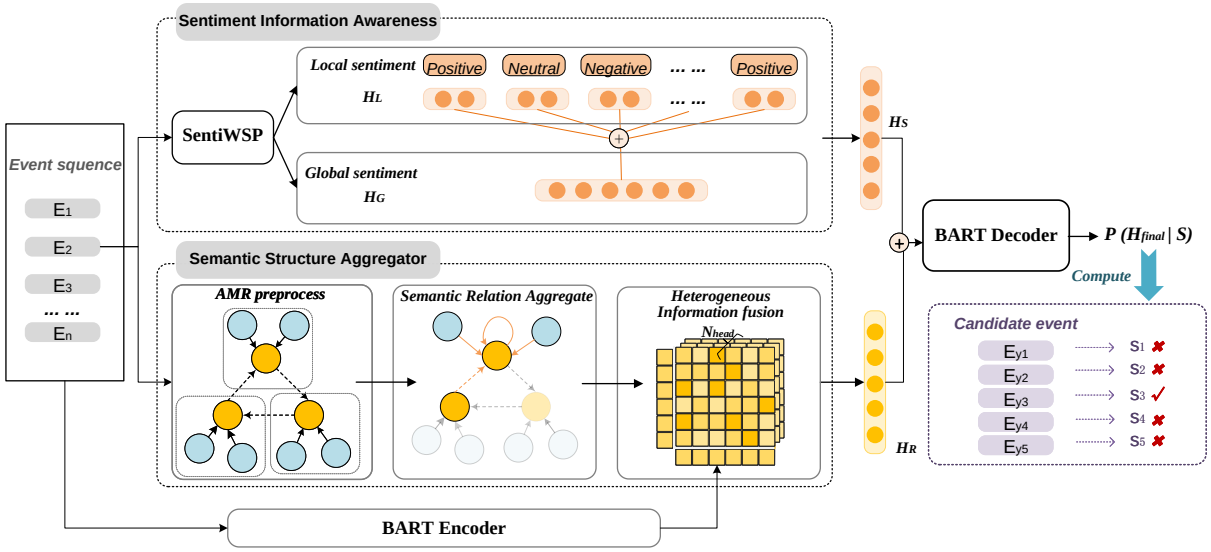

Figure 3: An illustration diagram of the proposed model.

neighborhood information, capturing complex patterns in the graph's relational structure. The final node representations are $H_A = \{h_1, h_2, ..., h_n\}$, where n is the number of nodes.

2.2.3 Heterogeneous Information fusion

In order to better integrate information from different sources, SSD-GM takes the original event representation H and the H_A derived from AMR structures into the conventional attention mechanism. This integration is articulated in the following formulation:

$$A_R = \text{softmax}\left(\frac{HW_Q \times (H_A W_K)^T}{\sqrt{d_w}}\right)$$

Here, the input vectors H and H_A are input vectors with d dimensions, while W_Q and W_K are learnable weights, with the same size of $R^{d\times d}$. With H_A , attention outputs are further guided by the semantic information from AMRs, which improves the efficient attention to semantic keywords.

In addition, we introduced the gating mechanism into the semantic enhanced self-attention, as follows:

$$G = \text{sigmoid}(HW_G)$$

$$H_R = (HW_V)A_R \odot G$$

In these expressions, W_G and W_V represent sets of trainable parameters, and G functions as a gating matrix. Given that a minority of words within a sentence typically carry significant meaning, this gating mechanism is instrumental in filtering out irrelevant background noise. This, in turn, facilitates the model's ability to concentrate on the more pivotal words, thereby enhancing the overall effectiveness of the attention mechanism.

Finally, with all these above calculations including, we obtain the semantic-enhanced representation $H_R = \{h_1, h_2, \ldots, h_n\}$ for subsequent task.

2.3 Sentiment Information Awareness

To enhance the effectiveness of event sentiment in differentiating between similar event pairs, we propose the development of a Sentiment Awareness Module designed to capture both global and local sentiments associated with events.

2.3.1 Local Sentiment

Local sentiment involves mapping the emotional reactions of participants to vector representations, thus highlighting how individuals perceive and respond to events. This method helps address semantic biases, allowing for a more accurate distinction between events that, despite sharing similar wording, convey very different emotional undertones. For instance, consider the statements "Bob gives an apology" and "Bob gives praise." Although the structure of the two events might appear alike, the emotions they trigger in Bob are quite distinct. After "Bob gives an apology," he is likely to feel regret or remorse, while after "Bob gives praise," he would likely experience pride or satisfaction. These emotional subtleties are key to differentiating between events that may seem superficially similar but are emotionally divergent. We use SentiWSP (Fan et al., 2022), which translates these event words (W = w_1, w_2, ..., w_n) into binary sentiment classifications: positive (labeled as 1), neutral (labeled as 0), or negative (labeled as -1). The local sentiment polarity of an event p_i^l is then

determined via SentiWSP. The sentiment labels for each event are concatenated to get embedding $H_L = \{p_1^l, p_2^l, ..., p_n^l\}$.

2.3.2 Global Sentiment

Global sentiment refers to the overall emotional trajectory across a sequence of events. This holistic perspective on emotional consistency can also play a significant role in predicting subsequent events within a given task. Likewise, we use SentiWSP, which is also skilled at capturing the emotional information in the long term to obtain a global sentiment label embedding H_G for the entire sequence of events.

2.4 Auto-regressive Generative Inference

We combine the representations derived from the three components to form the final event sequence representation as the BART decoder input, which is denoted as H_{final} and is achieved by concatenating the separate representations: H_R, H_L, and H_G.

With the auto-regressive generative model BART, the output embedding (last hidden states) from the BART decoder is the conditional probability distribution $P(H_{final} \mid S)$.

According to $P(H_{final} \mid S)$, each event candidate y_i calculates the probability $P(y_i \mid X)$. Since each event candidate has a different number of tokens, to avoid the model favoring shorter event candidates, we compute the mean of the log-probabilities of the verbalizer tokens for each event as the score o_i for the event candidate y_i. Specifically:

$$o_i = \frac{1}{N_{y_i}} \sum_{n=2}^{N_{y_i}} \log P_{\text{LM}}(y_n \mid X_m, y_{1:n-1})$$

where N_{y_i} is the number of tokens in the event candidate y_i. We then use the softmax function to calculate the final score s_i for each event candidate y_i:

$$s_i = \frac{\exp(o_i)}{\sum_{k=1}^{M} \exp(o_k)}$$

where M is the total number of event candidates. Finally, we define the loss function as follows:

$$L_{\text{cot}} = -\log(s_t) + \frac{1}{M-1} \sum_{\substack{i=1 \\ i \neq t}}^{M} \left(\frac{s_i}{1-s_t}\right) \log\left(\frac{s_i}{1-s_t}\right)$$

where t denotes the index of the correct event candidate y_t, and M is the number of event candidates. The first term of the loss function is the conventional softmax cross-entropy objective, which aims to maximize the probability of the correct event candidate. The second term, inspired by (Chen et al., 2019), maximizes the entropy of the probabilities for the negative event candidates conditioned on the correct event candidate y_t not occurring.

	Original Dataset	Public Dataset
Train set	1,440,295	40,331
Dev set	10,000	10,000
Test set	10,000	10,000

Table 2: The statistics of the reproduced original dataset(Granroth-Wilding and Clark, 2016) and the public dataset(Zhu et al., 2023).

3 Experiments

In this section, we introduce the datasets, experimental setting, and compared baselines. Experimental results show our method achieves state-of-the-art performance on the multichoice narrative cloze (MCNC) task. We then perform an ablation study and model training comparison to understand the effect of the model's key components and their variants on performance. Finally, we conduct a case study to show how our model predicts the subsequent event.

3.1 Datasets

We use two datasets to evaluate the proposed framework. Basic statistics of the two datasets are shown in Table 2.

MCNC public dataset: Li et al. (2018) extract event chains from the New York Times portion of the Gigaword corpus (Graff et al., 2003) following Granroth-Wilding and Clark (2016). Specifically, it uses news categorized as "story" from the year 1994 to 2004. It utilizes the C&C tool (Curran et al., 2007) for event extraction and OpenNLP 4 for coreference resolution.

MCNC original dataset: Zhu et al. (2023) reproduce the extraction of event chains and obtain a larger data set in the same way. They manually filter the extremely short event chains and truncate the long event chains so that the length of the remaining event chain (defined as script) is nine. The last event of the script is used for the positive event candidate. Negative event candidates are randomly sampled, where the protagonist is kept the same as the protagonist of the current script, and other arguments (object or indirect object) are replaced randomly by other arguments from the same document.

For both the "original" and "public" datasets, each instance has five event candidates, of which

only one choice is correct. To demonstrate the effectiveness of our proposed approach, we choose to evaluate our model on both.

3.2 Experimental Settings

To compare with baselines of different sizes of datasets, we conduct experiments on BART$_{base}$ and BART$_{large}$. The model is optimized by Adam (Kingma, 2014). The learning rate and weight decay are 1e-5 and 1e-6. Our model uses an early stop strategy to select the best epoch, with patience set to 5. For BART$_{base}$ and BART$_{large}$, the batch size is set as 64 and 32, respectively. All the experiments are conducted on NVIDIA GeForce RTX 4090. Accuracy is adopted as the evaluation metric.

3.3 Baselines

We apply the following representative methods as baselines. We divided them into the following categories:

Event Representations: 1) **Event-Comp** (Granroth-Wilding and Clark, 2016) uses the training objectives like Word2Vec (Mikolov et al., 2013) to learn event embeddings and calculates pairwise similarities between script events and event candidates. 2) **Pair-LSTM** (Wang et al., 2017) uses LSTM to model the narrative order of script events. 3) **SAM-Net** (Lv et al., 2019) uses LSTM and self-attention mechanism to capture diverse event segments. 4) **MCPredictor** (Bai et al., 2021) obtains event representations from pretrained Word2Vec and enhances them with original sentence representations obtained by pretrained BERT. Moreover, multiple similar event chains are utilized to aggregate script-level information to help select the subsequent event. 5) **SCPredictor-s** is an ablation of MCPredictor, removing additional similar scripts and the original sentence information. 6) **BART** (Lewis, 2019) fine-tunes the pre-trained model BART with a linear classifier. 7) **Two-stage BART** (Zhu et al., 2023) was trained in two phases, using task-centered and contrast fine-tuning.

Methods Enhanced with Structured Information: 1) **RoBERTa + Know. Model** (Zhou et al., 2021) learns a knowledge model from ASER to predict event relations. 2) **SGNN** (Li et al., 2018) constructs a narrative event evolution graph via verb con-occurrence frequency to obtain more effective event representations. 3) **GraphBERT** (Du et al., 2022) builds an event graph similar to SGNN and enhances BERT with the event graph. 4) **SCpredictor** (Bai et al., 2021) explores the original sentences of each event to enhance SCPredictor-s. 5) **ASER-Enhancement** (Zhou et al., 2021) incorporates eventuality knowledge graph ASER (activities, states, events, and their relations) to predict the subsequent event. 6) **REP** (Bai et al., 2023) explores the rich event description parsed from Abstract Meaning Representation (AMR) to boost the event prediction. 7) **EventBERT** (Zhou et al., 2022a) pretrains RoBERTa on BOOKCORPUS (Zhu et al. 2015) with three self-supervised contrastive learning objectives. 8) **ClarET** (Zhou et al., 2022b) pretrains BART on BOOKCORPUS with three additional self-supervised objectives.

Method	Accuracy(%)	Category
EventComp	49.57	w/o ext.
PairLSTM	50.83	w/o ext.
SGNN	52.45	w/o ext.
SAM-Net	54.48	w/o ext.
GraphBERT	60.72	w/o ext.
RoBERTa$_{large}$ + Know. Model	59.99	w/ ext.
ASER Enhancement	58.66	w/ ext.
BART$_{base}$	60.00	w/o ext.
Two-stage BART$_{base}$	62.54	w/o ext.
SSD-GM (BART$_{base}$)	63.91	w/ ext.

Table 3: Base model accuracy on the test set of the public dataset. "ext." is short for external knowledge.

Method	Accuracy(%)	Category
EventBERT	63.50	w/ ext.
RoBERTa$_{large}$ + Know. Model	63.62	w/ ext.
ClarET	64.61	w/ ext.
Two-stage BART$_{large}$	64.82	w/o ext.
SSD-GM (BART$_{large}$)	65.19	w/ ext.

Table 4: Large model accuracy on the test set of the public dataset.

3.4 Overall Performance Comparison

We first present the main experimental results of the widely-used "public" dataset in Table 3 and Table 4 using the base and large model, respectively, in order to align with the existing baselines for comparable parameters. Though prior researchers mentioned that a 1% improvement in accuracy for SEP is challenging. We can draw the following observations from the results on two model settings: 1). Our approach achieves the new state-of-the-art performance with a comparable amount of parameters and obtains a 1.37% improvement over the best baseline, Two-stage BART. Two-stage BART is modeled by a two-stage fine-tuned BART model; however, like traditional pre-trained language models, Two-stage BART ignores the importance of semantic structure for event representation and distinguishing between embeddings of similar events. 2). Moreover, our approach even outperforms strong

baselines that perform heavy event-centric post-pretraining such as ClarET and Two-stage BART. It demonstrates that more extensive semantic structure information can lead to more precise event representation knowledge.

Then, we evaluate our approach on the original dataset, present the experimental results in Table 4, and observe the following key findings: 1). Our method still achieves comparable performance to MCPredictor and Two-stage BART, although MCPredictor improves performance by 8.35% with additional original sentence and multi-script knowledge aggregation, which is also inconvenient for applying the model to downstream tasks. 2). It is noteworthy that methods employing direct representations of events using AMR exhibit relatively poorer performance. This is attributed to the variability in the number of event arguments, which hampers the training efficacy of predictive models.

Method	Accuracy(%)	Category
EventComp	50.19	w/o ext.
PairLSTM	50.32	w/o ext.
SGNN	52.30	w/o ext.
SAM-Net	55.60	w/o ext.
SCPredictor-s	58.79	w/o ext.
SCPredictor	66.24	w/ ext.
MCPredictor	67.05	w/ ext.
REP	60.08	w/ ext.
Two-stage BART$_{base}$	67.21	w/o ext.
SSD-GM (BART$_{base}$)	68.32	w/ ext.

Table 5: Base model accuracy on the test set of the original dataset.

3.5 Comparison with Large Language Models

We observe that generative LLMs like ChatGPT [1], Claude-3-5-sonnet and Llama 3.1 70b are challenged for script event prediction tasks in the zero-shot setting. We adopted an in-context learning approach for zero-shot tasks and designed a prompt specifically tailored for the script event prediction task, as shown in Table 6. As shown in Table 6, GPT-3.5 and Llama 3.1 70B, with accuracy scores of 28.36% and 39.32%, demonstrate comparable difficulties in modeling event dependencies. These models' performance underscores the challenges LLMs face in capturing the detailed and dynamic nature of event interactions. Claude-3-5-sonnet and GPT-4o, while showing some level of improvement over earlier models, still fall short in their ability to model the nuanced interactions between events. Smaller supervised models perform better than zero-shot LLMs in this domain, which can be attributed to their challenges in effectively modeling the intricate relationships between events.

[1] https://openai.com/research/gpt-4

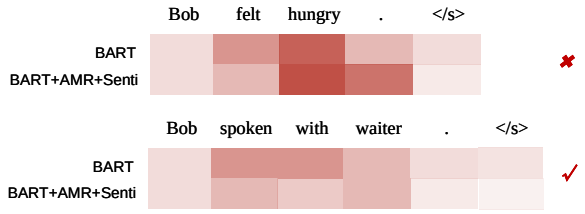

Figure 4: Visualizing the impact of AMR and sentiment for generative model.

3.6 Ablation Study

Table 7 shows our ablation analysis for our approach. We first investigated the impact of the semantic structure aggregator (row1) and found that the model's performance dropped by 2.59% without this stage. This is because the model can leverage semantic structural information to resolve ambiguities in event semantics. Enhanced quality in event representation enables the model to learn more comprehensive and nuanced knowledge. Second, when we remove our sentiment information awareness (row2), we observe that the sentiment information awareness is crucial because it enables the model to distinguish between similar event pairs and predict subsequent events within a given emotion topic. Moreover, it is better to aggregate the semantic structure relation in our method instead of the GCN and CompGCN (Vashishth et al., 2019) (row3, row4) methods due to the advantages of polymerized heterogeneous edges. Finally, we explore the importance of the two types of sentiment information. We find that our introduced local sentiment information (row5) is slightly better than the global sentiment information (row6), which shows the importance of the distinction between similar events for the task.

3.7 Case Study

We conducted a case study in our dataset to demonstrate the importance of aggregating AMR semantic structure information and sentiment information for our model, and to illustrate how SSD-GM predicts correct answers. As shown in the Figure 4, SSD-GM generates a score for each token in a candidate event in an autoregressive method, with lower scores indicating higher generation probabilities. Compared to the traditional model, the lower the generation probability of SSD-GM for wrong candidate events, the higher the generation probability of SSD-GM for correct answers.

Component	Description
Task Description	Assume you are a script event prediction classifier. Given a script event chain composed of 8 events ordered chronologically and a candidate event set composed of 5 candidate events, you need to select the most likely next event from the candidate event list.
Input Format	A script event chain consisting of 8 events, ordered chronologically, and a candidate event set of 5 candidate events.
Output Format	The output must be only one character: A, B, C, D, or E, corresponding to the most likely next event.
Constraints	The output is restricted to one character only. No additional explanation or text should be provided.

Table 6: Prompt Design for Script Event Prediction

Methods	Public dataset	Original dataset
SSD-GM (BART$_{base}$)	63.91	68.32
GPT-3.5	28.36	28.71
GPT-4o	47.97	48.21
Claude-3-5-sonnet	47.24	47.40
Llama 3.1 70B	39.32	43.69

Table 7: Comparison of accuracy with Large Language Models on the test set in public Dataset

Method	Accuracy(%)
SSD-GM (BART$_{base}$)	63.91
w/o semantic structure aggregator	61.32
w/o sentiment information awareness	62.42
replace with GCN	62.89
replace with CompGCN	63.39
w/o global sentiment information	63.55
w/o local sentiment information	62.98

Table 8: Ablation study of our SSD-GM model accuracy on the test set of the public dataset.

4 Related Work

Script event prediction was introduced by Chambers and Jurafsky (2008) through the narrative cloze task, where a missing event in a narrative is predicted based on co-occurrence frequencies. In this context, an event is defined as a verb and its dependency. Granroth-Wilding and Clark (2016) expanded this by defining events as verbs and their arguments (subject, object, indirect object) and proposed the MCNC task, which selects the next event from a set of candidates based on the narrative chain. The MCNC task has since become the standard benchmark for evaluating script event prediction models.

Early approaches, such as Granroth-Wilding and Clark (2016), employed Word2Vec to represent events and inferred probabilities based on similarities but overlooked the narrative order. Later studies (Wang et al., 2017; Lv et al., 2019) addressed this by integrating LSTM networks to capture the narrative sequence. Progress in understanding event relations has taken two primary directions. The first employs graph-based methods: Li et al. (2018) created a narrative event graph using verb co-occurrence and applied graph neural networks for enhanced event representation. Gao et al. (2022) introduced contrastive learning and clustering techniques, while Du et al. (2022) combined BERT with graph neural networks to incorporate event graph information. The second direction uses discourse-based methods: Lee et al. (2020) constructed a narrative event graph by extracting discourse relations and reformulated the task as link prediction. Bai et al. (2021) and others (Lv et al., 2020) enhanced event representations by integrating knowledge graphs. However, discourse-based methods are often hindered by limited relation types and sparse event graphs.

5 Conclusion

In this paper, we address the challenges in Script Event Prediction (SEP), which involves forecasting the next event in a given sequence from a set of candidates. Traditional methods struggle with event semantic ambiguity and embedding bias—issues arising from varied interpretations of similar vocabulary and the indistinguishable embeddings of related event pairs. To overcome these challenges, we propose a novel model integrating Semantic Structure and Sentiment Awareness (SSD-GM). Our approach begins with the Semantic Structure Aggregator, which enhances event representation using AMR. We then introduce an AMR-based semantic relation integration module to extract event-centric structural information and develop a heterogeneous information fusion mechanism to refine event features. Additionally, we incorporate a Sentiment Information Awareness Module to capture both

global and local sentiments of events, improving the differentiation between similar event pairs. Extensive experiments on the multi-choice narrative cloze (MCNC) task show that our SSD-GM model significantly outperforms existing state-of-the-art methods.

6 Acknowledgments

We would like to thank the PC chairs and anonymous reviewers for their invaluable suggestions and feedback. This work was partially supported by NSFC (Nos. U23A20296, 62272469, 72371245, 72471237), and the Science and Technology Innovation Program of Hunan Province (No. 2023RC1007).

7 Limitations

Despite the advancements presented in our article, there are notable limitations that should be acknowledged. One significant limitation is the reliance on AMR for enhancing event representation, which may not fully capture the dynamic and contextual variations of events across diverse narratives. Additionally, the effectiveness of the Sentiment Information Awareness Module, while promising, is contingent on the quality and granularity of sentiment analysis, which can be challenging in cases of nuanced or ambiguous emotional content. Furthermore, our experiments were conducted primarily on the multi-choice narrative cloze (MCNC) task, which may not generalize across all script event prediction scenarios or other types of event-centric tasks. Future work should explore these aspects further and test the model's applicability to a broader range of datasets and tasks to validate its robustness and adaptability.

References

Long Bai, Saiping Guan, Jiafeng Guo, Zixuan Li, Xiaolong Jin, and Xueqi Cheng. 2021. Integrating deep event-level and script-level information for script event prediction. *arXiv preprint arXiv:2110.15706*.

Long Bai, Saiping Guan, Zixuan Li, Jiafeng Guo, Xiaolong Jin, and Xueqi Cheng. 2023. Rich event modeling for script event prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12553–12561.

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.

Austin Blodgett and Nathan Schneider. 2021. Probabilistic, structure-aware algorithms for improved variety, accuracy, and coverage of amr alignments. *arXiv preprint arXiv:2106.06002*.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797, Columbus, Ohio. Association for Computational Linguistics.

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.

Hao-Yun Chen, Pei-Hsin Wang, Chun-Hao Liu, Shih-Chieh Chang, Jia-Yu Pan, Yu-Ting Chen, Wei Wei, and Da-Cheng Juan. 2019. Complement objective training. *arXiv preprint arXiv:1903.01182*.

James R Curran, Stephen Clark, and Johan Bos. 2007. Linguistically motivated large-scale nlp with c&c and boxer. In *Proceedings of the 45th annual meeting of the Association for Computational Linguistics Companion volume proceedings of the demo and poster sessions*, pages 33–36.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Du, Xiao Ding, Yue Zhang, Kai Xiong, Ting Liu, and Bing Qin. 2022. A graph enhanced bert model for event prediction. *arXiv preprint arXiv:2205.10822*.

Shuai Fan, Chen Lin, Haonan Li, Zhenghao Lin, Jinsong Su, Hang Zhang, Yeyun Gong, Jian Guo, and Nan Duan. 2022. Sentiment-aware word and sentence level pre-training for sentiment analysis. *arXiv preprint arXiv:2210.09803*.

Jun Gao, Wei Wang, Changlong Yu, Huan Zhao, Wilfred Ng, and Ruifeng Xu. 2022. Improving event representation via simultaneous weakly supervised contrastive learning and clustering. *arXiv preprint arXiv:2203.07633*.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

Mark Granroth-Wilding and Stephen Clark. 2016. What happens next? event prediction using a compositional neural network model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

I-Ta Lee and Dan Goldwasser. 2019. Multi-relational script learning for discourse relations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4214–4226.

I-Ta Lee, Maria Leonor Pacheco, and Dan Goldwasser. 2020. Weakly-supervised modeling of contextualized event embedding for discourse relations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4962–4972.

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

Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Constructing narrative event evolutionary graph for script event prediction. *arXiv preprint arXiv:1805.05081*.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Shangwen Lv, Wanhui Qian, Longtao Huang, Jizhong Han, and Songlin Hu. 2019. Sam-net: Integrating event-level and chain-level attentions to predict what happens next. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6802–6809.

Shangwen Lv, Fuqing Zhu, and Songlin Hu. 2020. Integrating external event knowledge for script learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 306–315.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, pages 593–607. Springer.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2019. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082*.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. 2018. R 3: Reinforced ranker-reader for open-domain question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Zhongqing Wang, Yue Zhang, and Ching Yun Chang. 2017. Integrating order information and event relation for script event prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 57–67.

Yucheng Zhou, Xiubo Geng, Tao Shen, Guodong Long, and Daxin Jiang. 2022a. Eventbert: A pre-trained model for event correlation reasoning. In *Proceedings of the ACM Web Conference 2022*, pages 850–859.

Yucheng Zhou, Xiubo Geng, Tao Shen, Jian Pei, Wenqiang Zhang, and Daxin Jiang. 2021. Modeling event-pair relations in external knowledge graphs for script reasoning. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Yucheng Zhou, Tao Shen, Xiubo Geng, Guodong Long, and Daxin Jiang. 2022b. Claret: Pre-training a correlation-aware context-to-event transformer for event-centric generation and classification. *arXiv preprint arXiv:2203.02225*.

Fangqi Zhu, Jun Gao, Changlong Yu, Wei Wang, Chen Xu, Xin Mu, Min Yang, and Ruifeng Xu. 2023. A generative approach for script event prediction via contrastive fine-tuning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 14056–14064.