Incorporating Circumstances into Narrative Event Prediction

Shichao Wang[1,2], Xiangrui Cai[1,2,*], Hongbin Wang[3], Xiaojie Yuan[2,4]
[1]College of Cyber Science, Nankai University, Tianjin, China
[2]Tianjin Key Laboratory of Network and Data Security Technology, China
[3]Mashang Consumer Finanace Co, Ltd
[4]College of Computer Science, Nankai University, Tianjin, China
wangshichao@dbis.nankai.edu.cn
{caixr,yuanxj}@nankai.edu.cn
hongbin.wang03@msxf.com

Abstract

Aiming at discovering the event evolution, the narrative event prediction is essential to modeling sophisticated real-world events. Existing studies focus on mining the inter-events relationships while ignoring how the events happened, which we called *circumstances*. However, we observe that the circumstances indicate the event evolution implicitly, and are significant for the narrative event prediction. To incorporate circumstances into the narrative event prediction, we propose the **CircEvent**, which adopts the multi-head attention to retrieve circumstances at the local and global levels. We also introduce a regularization of attention weights to leverage the alignment between events and local circumstances. The experimental results demonstrate that CircEvent outperforms existing baselines by 12.2%. Further analysis demonstrates the effectiveness of our multi-head attention modules and regularization. Our source code is available at https://github.com/Shichao-Wang/CircEvent.

1 Introduction

The Narrative event chain, which is similar to the classical notion of the *script* (Schank and Abelson, 2013), is a structural knowledge that captures the relationships between event sequences and their participants in the given scenario. Figure 1 describes a scenario of "going to the restaurant.". Modeling the narrative event chain can help the AI systems to understand sophisticated real-world events and benefit many downstream applications (Han et al., 2021), such as financial analysis (Yang et al., 2019). This paper focuses on modeling the narrative event chain and predicting what will happen next, which is called the Multiple Choice Narrative Cloze (MCNC) (Granroth-Wilding and Clark, 2016). As shown in Figure 1, the MCNC evaluation

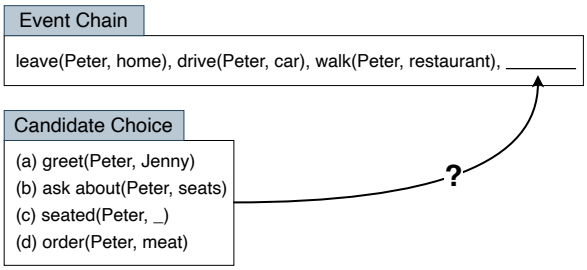


Figure 1: An example of multiple choice narrative cloze (MCNC). It aims to predict the correct next event from the candidates events given context events.

aims to choose the correct event from the candidate choices set given a sequence of historical events.

Early studies learn event representation with the rule-based (Schank and Abelson, 2013), count-based (Chambers and Jurafsky, 2008; Pichotta and Mooney, 2016), and deep learning (Modi and Titov, 2014a,b; Granroth-Wilding and Clark, 2016) method. Recently, more and more studies attempt to incorporating external knowledge into event representation. Li et al. (2018) builds the Narrative Event Evolutionary Graph (NEEG), which describes event evolutionary principles and patterns. FEEL (Lee and Goldwasser, 2018) introduces a feature enriched event embedding. Despite the subject, predicate, and object, FEEL also considers sentiment and animacy as the parts of events. Lee and Goldwasser (2019) regards event embedding learning as a multi-relational problem and captures different relations of events pairs, such as the cause and contrast. Zheng et al. (2020b) builds a heterogeneous event graph to mining subordinated relations between events and words.

In addition to the previous events that have already happened, particular situations also affect the event evolution, which are defined by *circumstances* in this paper. The event circumstances include detailed descriptions of the event situation such as the weather, the place status, and the protagonist behavior. As the example shown in Figure 1,

*Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4840–4849
November 7–11, 2021.

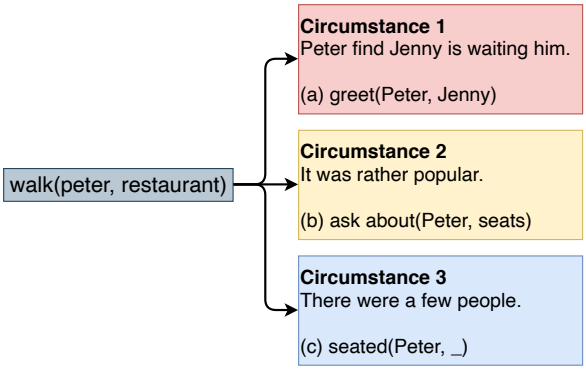


Figure 2: Examples of event circumstance. Different circumstance is boxed in different color, and the possible next event is placed at the bottom of the box.

existing works tend to predict choice (c) or (d) based on previous events or historical knowledge for the event *walk (Peter, restaurant)*. Given different circumstances shown in Figure 2, they could not make a different decision based on the restaurant's environment, such as described in circumstances 2 and 3. Peter is more likely to get seated if the restaurant has a few customers. He will ask the waiter about the available seats if the restaurant is popular, meaning it is crowded. Circumstances, such as the crowdedness of the restaurant, weather, or the protagonist action et al., will influence the event evolutionary, while existing works do not consider them.

In this paper, we propose CircEvent to represent events together with their circumstances. Following previous studies (Chambers and Jurafsky, 2008; Lee and Goldwasser, 2019), events in this paper are also extracted from the unstructured text corpus, and each event belongs to one specific sentence in the text. The extracted event only contains the minimum information of an event, e.g., the subject, predicate, and object. The contextual information, environment description, and semantics, which we discussed as *circumstances*, are left in the original sentence. We attempt to collect event circumstance information from the unstructured text. However, the unstructured text contains so much information that not all contribute to the event evolution.

To tackle this challenge, we develop two multi-head attention-based networks to incorporate event representation and its circumstance into narrative event prediction at the *local* and *global* levels. At the *local* level, events come from a specific sentence, containing the most related circumstance. We develop the a multi-head attention to retrieve the local circumstances for events. Moreover, the context local circumstances also contribute the event representation. We develop another multi-head attention to get the global circumstances by aggregating the context local circumstances adaptively.

After the circumstances retrieval, we adopt the transformer as backbone to encode the context events and circumstances. The transformer decoder is used to compute the similarity scores of candidate events. The candidate events are compared implicitly inner the transformer decoder benefited from its architecture.

Our contributions in this paper are three folds:

1. We propose the CircEvent to incorporate event circumstances into narrative event prediction with the transformer architecture.
2. We introduce two multi-head attention to retrieve the event circumstances from the corpus at the local and global levels.
3. Our proposal outperforms the existing baselines by 12.2% on the MCNC task, and our further analysis proves the effectiveness of event circumstances.

2 Related Work

2.1 Narrative Event Representation

In the literature, the methods to get event representation can be categorized in two: the *self-contained* and the *external knowledge enriched*.

In the *self-contained* event representation research work, they only use the events and corresponding connection relation. Event-Comp (Granroth-Wilding and Clark, 2016) employed distributed representation, word2vec (Mikolov et al., 2013), to learn the word representation of arguments that appear in the event, and the event representation is the linear combination of these arguments representation. RoleFactor (Weber et al., 2018) proposed a scalable tensor-based composition model for event representations, which composite event argument in a hierarchical structure. The UniFA-S (Zheng et al., 2020a) adopted Variational AutoEncoder architecture(Kingma and Welling, 2014) with a unified fine-tuning method to learn event representation from intra- inter-event and scenario level. HeterEvent (Zheng et al., 2020b) proposed a heterogeneous graph neural network that models discontinuous event segments explicitly.

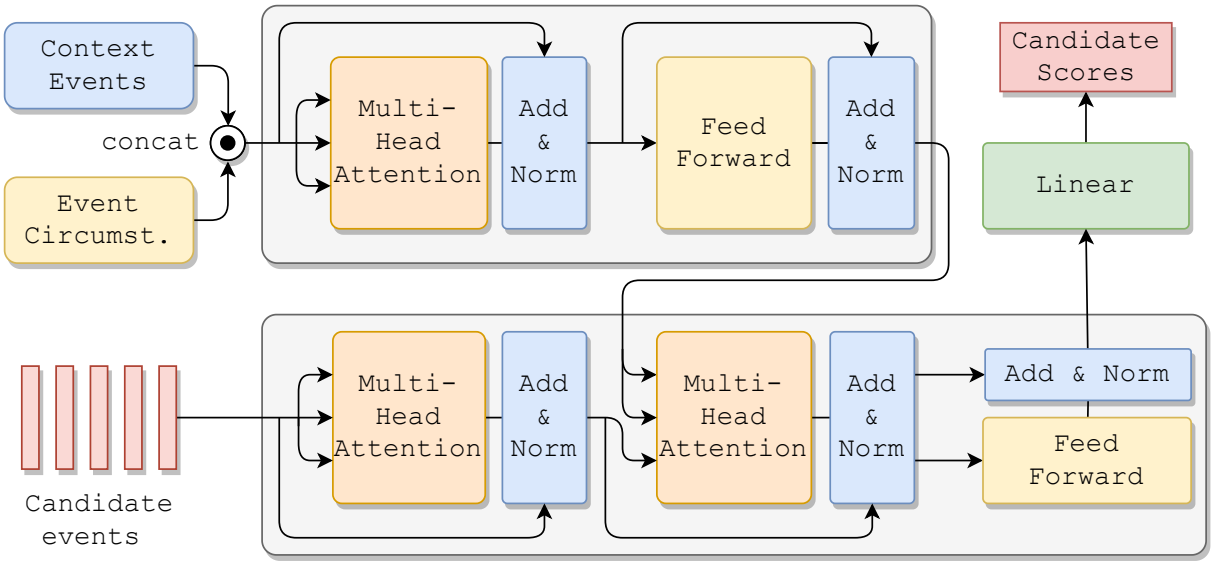


Figure 3: The architecture of CircEvent. The context events and circumstances are concatenated and fed into the transformer encoder, at the top-left corner. The similarity scores of candidate events are output by the transformer decoder with a linear pooling layer at the bottom-right. The event circumstance representation method is detailed in Sec 3.2.

In the *external knowledge enriched* representation research line, researchers have attempted several external knowledge into event representation. FEEL (Lee and Goldwasser, 2018) injects sentiment and animacy information into event embedding. (Yang et al., 2019) enrich event representation with news information. EventTransE (Lee and Goldwasser, 2019) regards event embedding learning as a multi-relational problem and incorporates relationships among events into event representation learning. (Ding et al., 2019) leverage commonsense knowledge about intent and sentiment into the event, which can be found in the knowledge bases such as Event2Mind (Rashkin et al., 2018) and ATOMIC (Sap et al., 2019). In this paper, we attempt to incorporating event circumstances into event representation.

2.2 Attention Mechanism in Narrative Event Prediction

Since (Bahdanau et al., 2015) firstly adopt attention mechanism in neural machine translation. The attention mechanism has shown its effectiveness in many NLP applications. Many previous works on the narrative event prediction (Wang et al., 2017; Li et al., 2018; Lv et al., 2019) also apply attention mechanism to the context events, as they assume different context events have different weights for choosing the correct subsequent event. Besides, Lv et al. (2019) employs a self-attention mechanism (Lin et al., 2017) to represent the event chain in diverse event segments within the chain implicitly. Zheng et al. (2020b) adopt the graph attention network (Velickovic et al., 2018) to aggregate neighborhood events information. We employ the multi-head attention (Vaswani et al., 2017) to extract circumstance representation from the event sentence and aggregate circumstances in the global level adaptively.

3 Model

This section introduces our CircEvent neural network in four modules: the event representation, the circumstance representation, the event chain encoder and the prediction module.

3.1 Event Representation

Each event consists of three arguments, i.e., subject, predicate, and object. Each argument has n_{args} words. We follow (Zheng et al., 2020b) to apply a max-pooling and an average pooling layer on argument word embeddings and then concatenate them to get the event argument embeddings. The subject, predicate and the object representaion are denoted as $s(e), p(e), o(e) \in \mathbb{R}^{2d_e}$. For the subject, its representation $s(e)$ follows:

$$s(e) = [\max(w_s); \text{avg}(w_s)]$$

where $w_s \in \mathbb{R}^{n_{arg} \times d_h}$ is the sequence of subject word embeddings. The max, avg, and $[;]$ refer to

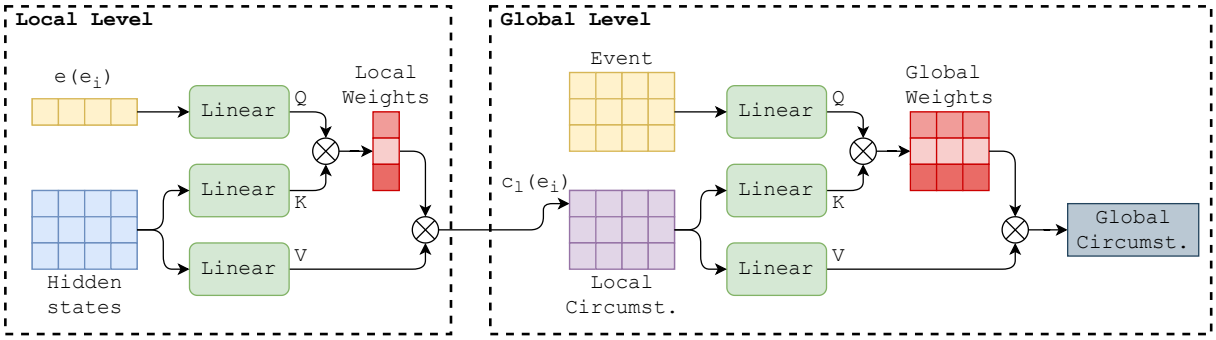


Figure 4: The circumstance representation module. The local circumstance is placed at the left side. The vector in yellow represent a single event embedding, and the matrix in blue is the corresponding sentence hidden states. The global circumstance is at right side. The event matrix in yellow is the context event embeddings, and the purple one is the local circumstances. Our regularization is applied on the Global Weights matrix.

the max pooling, average pooling and concatenation operation respectively. The predicate and the subject representation are obtained similarly.

The event embedding $e(e)$ is the linear combination of its argument vectors. Formally, the event embedding $e(e)$ definition follows:

$$e(e) = g(W_s s(e) + W_p p(e) + W_o o(e) + b)$$

$W_s, W_p, W_o \in \mathbb{R}^{d_h \times 2d_e}$, and $b \in \mathbb{R}^{d_h}$ are learnable parameters in our model. $g(\cdot)$ is a non-linear function, we employ a dense layer followed by a $\tanh(\cdot)$ activation here.

3.2 Event Circumstance Representation

This section details three methods to get the event circumstance representation $c(e)$.

Local As we described in Section 1, the event circumstance can be extracted from the sentence that contains the event. We first adopt a bidirectional recurrent neural network (BRNN) (Schuster and Paliwal, 1997) to retrieve the contextualized sentence hidden states. We adopt the multi-head attention (Vaswani et al., 2017) to aggregate sentence hidden states by corresponding event representation.

Suppose we have a sentence $s = [s_1, \ldots, s_{n_s}]$, which has n_s tokens, represented in the word embedding sequence. s_i is the word embedding vector of i^{th} token in the sentence. To equip each word with context information, we use bidirectional recurrent neural network to encode word embeddings to hidden states:

$$\overrightarrow{h_i} = \overrightarrow{\text{RNN}}(s_i, \overrightarrow{h_{i-1}})$$
$$\overleftarrow{h_i} = \overleftarrow{\text{RNN}}(s_i, \overleftarrow{h_{i+1}})$$

We use LSTM (Sak et al., 2014) as our recurrent neural network (RNN). $\overrightarrow{h_i}$ is the i^{th} word forward hidden state, and $\overleftarrow{h_i}$ is the backward. The forward and backward hidden state are concatenated to form the output hidden state $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}] \in \mathbb{R}^{d_h}$. We stack all the output hidden states to get the hidden states matrix $H = [h_1, h_2, \ldots, h_{n_s}] \in \mathbb{R}^{n_s \times d_h}$.

Inspired by (Vaswani et al., 2017), we use event embedding to query circumstances from contextualized sentence hidden states. The multi-head attention follows:

$$\text{head}_i = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_h}}\right) V_i$$
$$\text{MultiHead}(Q, K, V) = [\text{head}_1; \ldots; \text{head}_{n_h}] W^O \quad (1)$$

where $Q \in \mathbb{R}^{n_q \times d_h}$, $K \in \mathbb{R}^{n_l \times d_h}$, $V \in \mathbb{R}^{n_l \times d_h}$, $W^O \in \mathbb{R}^{n_h \cdot d_h \times d_h}$ are learnable parameters. n_q refers to query sequence length, n_l refers to context sequence length. The local event circumstance representation follows:

$$c_l(e) = \text{MultiHead}(e(e)W_l^Q, HW_l^K, HW_l^V)$$

where $W_l^Q, W_l^K, W_l^V \in \mathbb{R}^{d_h \times d_h}$. H is the hidden states matrix of the corresponding event sentence. Thus the local circumstance is highly related to the event. We can use that local circumstance embedding as the final circumstance embedding.

Global Instead of limited in the local circumstance, we claim that the context event circumstances also contribute to the event. We adopt another multi-head attention module to obtain the global circumstance from local circumstances.

The global circumstance computation follows Eq. 1 but with different Q, K, V. The global cir-

cumstance embedding follows:

$$c_g(e) = \text{MultiHead}(EW_g^Q, C_l W_g^K, C_l W_g^V)$$

where $E \in \mathbb{R}^{n \times d_h}$ is the sequence of context event embeddings. $C_l = [c_l(e_1), c_l(e_2), \ldots, c_l(e_n)] \in \mathbb{R}^{n \times d_h}$ is the sequence of context event local circumstances. Under the global circumstance setting, the contribution weights of context circumstances are obtained via the attention mechanism. However, collecting information from complicated context events with an attention mechanism is not reliable and ignores the alignment between events and sentences. Based on this assumption, we equip the global circumstance with a regularization.

Global + Regularization We observe that each event belongs to a specific sentence, and a sentence may contain multiple events. This aligned information can be formulated into a binary matrix Y. If the continuous events from i to j belongs to the same sentence, the sub-matrix $Y_{i:i+j,i:i+j}$ are set to 1. Figure 6c is a visual example, in which event 2, 3 and event 4, 5 belong to the same sentences respectively. We apply a regularized method on the attention heads, leading the event to aggregate more local circumstances of homologous events, which is extracted from the same sentences. The regularization computations follow:

$$L(A) = -\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(Y_{i,j}\log A_{i,j} + (1 - Y_{i,j})\log(1 - A_{i,j}))$$

$$A = \frac{1}{n_h}\sum_{i=1}^{n_h}\text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_h}}\right)$$

where $A_{i,j}$ is the average attention weights of i^{th} event to the j^{th} sentence. The n_h is the number of heads in the multi-head attention.

3.3 Event Chain Encoder

The encoder network structure refers to the Transformer Encoder proposed in (Vaswani et al., 2017). In the encoder, the event embedding and the circumstance embedding are concatenated to form the encoder inputs $x = [e(e); c(e)]$. The circumstance embedding $c(e)$ can be either $c_l(e)$ or $c_g(e)$. The formal formula follows:

$$C = \text{Encoder}([x_1, x_2, \ldots, x_n])$$

where x_i is the i^{th} composed embedding. $C \in \mathbb{R}^{n \times d_h}$ is the representation of the event chain.

3.4 Prediction Layer

We adopt the Transformer Decoder (Vaswani et al., 2017) as our prediction layer. The candidate events are fed into the decoder as queries. The transformer decoder contrasts the candidates events based on the event chain context. We adopt a dense layer to pool out the similarity scores of candidate events. The prediction layer follows:

$$O = \text{Decoder}([e(c_1), \ldots, e(c_{n_c})], C)$$
$$[s_1, \ldots, s_{n_c}] = w_o O^T + b$$

where c_i is the i^{th} candidate event, and $e(c_i)$ is the i^{th} candidate event embedding. The s_i is the similarity score of i^{th} candidate event, and the $w_o \in \mathbb{R}^{d_h}$ is a learnable vector.

We select the event choice with the highest score as the possible event to take place. There are five candidate events for each chain, and we apply $\text{softmax}(\cdot)$ to normalize and get the final score of each choice. We select the candidates with the maximum probability as the predicted event:

$$P(e_{c_i}|e_1, e_2, \ldots, e_n) = \frac{\exp(s_i)}{\sum_j \exp(s_j)}$$

where e_{c_i} is the i^{th} candidate event.

3.5 Training Object

Our main training object is to minimize the cross-entropy loss between the gold event and the predicted event. The main loss follows:

$$L(\Theta) = -\frac{1}{N}\sum_{k=1}^{N}\log P(e_{c_g}|e_1, e_2, \ldots, e_n) + \frac{\lambda}{2}||\Theta||_2^2$$

where Θ is the model parameters. The e_{c_g} refers to the ground truth event. The λ is the L2 regularization factor of model parameters. Moreover, Under the *Global + Reg* setting, the penalty on attention weights is also included with an α factor:

$$L = L(\Theta) + \alpha L(A)$$

4 Experiment

In this section, we describe the dataset and the preprocessing pipeline. We evaluate our model in the MCNC task and report the accuracy score.

	Train	Develop	Test
# Document	1,038,031	103,583	103,805
# Chain	419,106	52,328	52,811

Table 1: The statistic of NYT portion of Gigaword.

4.1 Dataset

Following Lee and Goldwasser (2019), we extract events from the NYT portion of the Gigaword corpus (Graff and Cieri, 2003). We use the Stanford CoreNLP (Manning et al., 2014) for POS tagging, dependency parsing, and coreference resolution. The extraction pipeline is detailed in the following paragraph. The event chains are split into the train set, develop set, and test set based on the documents split provided by Granroth-Wilding and Clark (2016). The detailed dataset statistics are shown in Table 1.

Event Chain Extraction In this paper, we describe an event in a triplet (*pred*, *subj*, *obj*), which means verb, subject, and object, respectively. We first use the POS tagger, dependency parser, and coreference resolver in Stanford CoreNLP (Manning et al., 2014) to annotate the raw corpus. Events are extracted following entities' coreference chain. We retrieve their predicate, subject, and object from the dependency parse tree for each mention in the coreference chain. We constraint the event arguments, e.g. subject, object, and predicate, length to $n_{arg} = 15$. For the sake of compatibility, we use a special token *UNK* for the missing arguments.

Take *Peter find Jenny is waiting him. He walks into the restaurant.* as an example. After the event extraction pipeline, there will have an event chain that contains *walk (Peter, restaurant)* and another contains *wait (Jenny, Peter)*.

Candidate Event Generator For each ground truth event, we follow (Lee and Goldwasser, 2019) to generate distractive events. We first collect all the events to construct an considerable event pool. An distract event is randomly sampled from the event pool, and then we randomly replace one of its arguments with that of the ground truth event. The ground truth and four distract events are combined and shuffled, serving as candidate events.

4.2 Baselines

We compare our model with following baselines:

- Event-Comp (Granroth-Wilding and Clark, 2016) is a neural network based on intra-events relationship.
- SGNN (Li et al., 2018) incorporates inter-events information by constructing a narrative event evolutionary graph (NEEG), which describes the event evolution patterns.
- SAM-Net (Lv et al., 2019) is an attention based model that captures event segments implicitly, and modeling the candidate events at the event-level and the chain-level.
- EventTransE (Lee and Goldwasser, 2019) is an representation learning method that explores discourse relations among events.
- HeterEvent$_{[W+E]}$ (Zheng et al., 2020b) is a representation learning method, which adopts heterogeneous event graph to capture the discontinuous event segments explicitly.
- UniFA-S (Zheng et al., 2020a) is a representation learning method based on the variational auto-encoder, which fine-tunes the pre-trained BERT (Devlin et al., 2019) on the NYT corpus and the event chains in multi-steps.
- SAM-Net$_{\text{Our}}$ We extend the SAM-Net to deal with the full event argument words rather than the headword and remove preposition from the event arguments for the comparability.

4.3 Experiment Configuration

The pre-trained Glove (Pennington et al., 2014) is used for word embedding, and the dimension d_e is set to 100. The input sentence length n_s is truncated to 60. For the transformer, the number of attention heads n_h is set to 4, and the number of encoder layer and decoder layer are set to 1. We set the batch size to 128. Adam (Kingma and Ba, 2015) is used to optimizing our model parameters. The learning rate set is to 1e-4, the λ is set to 1e-5, and the α is set ot 0.8. The model size d_h is set to 128. All the hyper-parameters are searched on validation set. The training process employs the early-stopping strategy on validation accuracy.

4.4 Results

We report the performance of the proposed CircEvent model and other baseline models on the NYT portion of the Gigaword corpus on the MCNC task in Table 2. The `CircEvent` shows outstanding performance and achieves the best accuracy score in the MCNC task.

We first zoom into the comparison among baselines. SAM-Net and our CircEvent are supervised learning methods in the narrative event prediction.

Method	Accuracy(%)
Event-Comp	46.3
SGNN	52.3
SAM-Net	54.3
EventTransE	63.7
HeterEvent$_{[W+E]}$	64.4
UniFA-S	66.3
SAM-Net$_{\text{Our}}$	72.4
CircEvent	**84.6**

Table 2: Performance on the MCNC test set. Our proposal CircEvent exceeds the best baselines by **12.2%**

Method	Accuracy(%)	Δ
Event + Global + Reg	84.64	-
Event + Global	83.60	- 1.04
Event + Local	84.32	- 0.32
Event	81.36	- 3.28
Global	81.51	- 3.13
Local	83.40	- 1.24

Table 3: Ablation studies on the event circumstances. The Event represents the event embedding, the Global and Local are the global and local circumstances, and Reg is the attention regularization.

However, the origin SAM-Net accepts the headword of event arguments, which lead to the corrupt event. Thus we re-implement it with the same representation layer use in CircEvent to solve the problems, serving as SAM-NET$_{\text{Our}}$, which performs the best of existing models.

Next, we compare our CircEvent with baselines. Our CircEvent achieves the best performance on the MCNC task, which is a definite 12.2% improvement over the best baselines. We discuss the contribution of each part further in Sec 4.5.

4.5 Ablation Study

In this part, we perform an ablation study to demonstrate the effectiveness of our neural network architecture. We depart our model in three parts, Event, Local, and Global refers to the event embedding, the local embedding, and the global embedding, respectively. We conduct the ablation studies based on them, and the results are shown in Table 3. Our ablation studies include the following two aspects:

Influence of Additional Circumstances We first study the influence of the additional local circumstances. We compare the result between the Event + Local and the Event. In the Event + Local experiment, the event embedding is concatenated with circumstance embedding. The event embedding is duplicated and concatenated in the Event experiment. The experiment results show that the local circumstances improve the accuracy score by 2.96%, which demonstrates the local circumstances containing valuable information to the next event.

Similar to the local circumstances, we compare the results between the Event + Global and the Event, which demonstrate that the additional global circumstance also contribute to the narrative event prediction. With the global circumstance embedding, the accuracy score increase by 2.24%. Comparing with the local and global circumstances, the local benefit more to the accuracy score. The attention matrix in global circumstance shows that all the context events relay on the last circumstance sentence most, because it is the closest one to the target event. We discuss it further in Sec 4.6.

Influence of Independent Circumstances The experiments above include event embedding. In this part, we would like to evaluate the quality of circumstance embedding independently. We remove the event embedding from the event chain encoder's input and use the local circumstances or the global circumstances to represent the event chain. The result is shown as Local and Global in Table 3. In these two experiments the event embedding is used as distantly supervised information, which aggregates the sentence hidden states and the local circumstances. From the result, we can conclude that the local and the global circumstance contains valuable information. With the distantly event embedding information, the Global and Local experiment results outperform that of the Event.

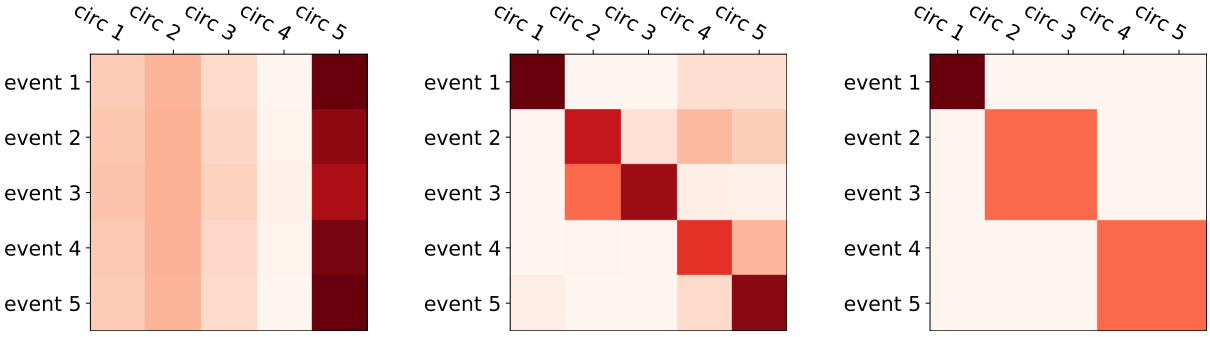
4.6 Qualitative Analysis

In this section, we provide a qualitative analysis of local circumstances and global circumstances.

Figure 5 is the visualization of local circumstance attention weights. Each row is a pair of the event and the sentence contains the event. The context events describe men were judged because of harvesting the abalone illegally. All of the last four events notice the word *abalone*, which is an important topic or element in the context but does not appear in the events. However, in the first sentence, the *abalone* is concatenated with the *May* creating an out-of-bag word. We blame this atten-

arrest (official, men)	State Fish and Game officials arrested the men with 468 red abaloneMay 20 when they landed Ward 's urchin boat .
admit (men, _)	The men admitted they harvested and planned to sell the abalone.
harvest (men, abalone)	The men admitted they harvested and planned to sell the abalone.
ban (judge, men)	A Mendocino County judge forever banned two SouthernCalifornia men from fishing , and sent them toprison for being caught with the largest single illegal abalone haulin California in 15 years .
send (judge, men)	A Mendocino County judge forever banned two SouthernCalifornia men from fishing , and sent them toprison for being caught with the largest single illegal abalone haulin California in 15 years .

Figure 5: The event chain and the local circumstance heat map example. The left side is the context events, which happen from the top to the bottom. The sentence that contains the event is placed directly right to the event. Words are wrapped in red color boxes. The deeper the shade of red, the more attention weight the word got.



(a) Un-regularized (b) Regularized (c) Target

Figure 6: The global attention weights matrices for the example given in Figure 5. The deeper the shade of red, the more attention it is. Figure 6a and Figure 6b show the attention weights without and with the regularization respectively. Figure 6c formulates the alignment between events and circumstances.

tion loss on the incorrect sentence tokenization. In the homologous events *admit (men, _)* and *harvest (men, abalone)*, which come from the same sentence, pay attention to the different parts of the sentence. The *harvest* event not only cares about the harvest predicate itself but also considers the coordinate verb *planned*. In the last two homologous events, the attention weights are incredibly similar. Despite the predicate itself, they also pay attention to the other words full of semantics, such as *largest, illegal*, and the topic *abalone*. Since, we use the linear combination to construct the event embedding, we think there is a lack of event expression that leads to similar heat maps.

We also visualize the global attention weights matrices in Figure 6. The attention weights lean to the last circumstance without our regularization. Thus, the Global and the Event + Global experiment results are worse than the Local and the Event + Local, respectively. With our regularization, the global attention weights align with the target matrix, which describes the alignment between the events and the circumstances. The leading diagonal elements have the prominent weight in each row in the regularized weights matrix. It means all the events pay attention mainly to their local circumstances. In the meantime, events also aggregate the information from context circumstances, especially the homologous events. The homologous events pay more attention to each other than to other events, such as the event 2, 3 and the event 4, 5 shown in Figure 6b. The results confirm our intuition that the circumstances have a significant influence on the narrative event prediction.

5 Conclusion

This paper develops multi-head attention modules to capture the circumstances from event text at local and global levels. We utilize the transformer architecture, encode context events and circumstances. The standard evaluation shows that our model achieves the best accuracy score compared to other baselines. The visual analysis on the attention heat map shows the effectiveness of circumstances.

Acknowledgements

We would like to thank all anonymous reviewers for their insightful comments. This research is supported by the NSFC-General Technology Joint Fund for Basic Research (No.U1936206, No.U1836109), the NSFC-Xinjiang Joint Fund (No.U1903128), the National Natural Science Foundation of China (No.62002178), and the Natural Science Foundation of Tianjin, China (No.20JCQNJC01730).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. *ICLR*.

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.

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

Xiao Ding, Kuo Liao, Ting Liu, Zhongyang Li, and Junwen Duan. 2019. Event Representation Learning Enhanced with External Commonsense Knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4894–4903, Hong Kong, China. Association for Computational Linguistics.

David Graff and Christopher Cieri. 2003. English Gigaword.

Mark Granroth-Wilding and Stephen Clark. 2016. What happens next? event prediction using a compositional neural network model. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2727–2733, Phoenix, Arizona. AAAI Press.

Yi Han, Linbo Qiao, Jianming Zheng, Hefeng Wu, Dongsheng Li, and Xiangke Liao. 2021. A survey of script learning. *Frontiers of Information Technology & Electronic Engineering*, 22(3):341–373.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. *ICLR*.

Diederik P. Kingma and M. Welling. 2014. Auto-Encoding Variational Bayes. *ICLR*.

I.-Ta Lee and Dan Goldwasser. 2018. FEEL: Featured Event Embedding Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

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, Florence, Italy. Association for Computational Linguistics.

Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Constructing narrative event evolutionary graph for script event prediction. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, pages 4201–4207, Stockholm, Sweden. AAAI Press.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-attentive Sentence Embedding. *arXiv:1703.03130 [cs]*.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6802–6809.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR*.

Ashutosh Modi and Ivan Titov. 2014a. Inducing Neural Models of Script Knowledge. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 49–57, Ann Arbor, Michigan. Association for Computational Linguistics.

Ashutosh Modi and Ivan Titov. 2014b. Learning Semantic Script Knowledge with Event Embeddings. *ICLR*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Karl Pichotta and Raymond Mooney. 2016. Statistical Script Learning with Recurrent Neural Networks. In *Proceedings of the Workshop on Uphill Battles in*

Language Processing: Scaling Early Achievements to Robust Methods, pages 11–16, Austin, TX. Association for Computational Linguistics.

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A. Smith, and Yejin Choi. 2018. Event2Mind: Commonsense Inference on Events, Intents, and Reactions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 463–473, Melbourne, Australia. Association for Computational Linguistics.

H. Sak, A. Senior, and F. Beaufays. 2014. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition. *ArXiv*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3027–3035.

Roger C. Schank and Robert P. Abelson. 2013. *Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures*. Psychology Press.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Petar Velickovic, Guillem Cucurull, A. Casanova, Adriana Romero, P. Lio', and Yoshua Bengio. 2018. Graph Attention Networks. *ICLR*.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Noah Weber, Niranjan Balasubramanian, and Nathanael Chambers. 2018. Event Representations With Tensor-Based Compositions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Yiying Yang, Zhongyu Wei, Qin Chen, and Libo Wu. 2019. Using External Knowledge for Financial Event Prediction Based on Graph Neural Networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, pages 2161–2164, New York, NY, USA. Association for Computing Machinery.

Jianming Zheng, Fei Cai, and Honghui Chen. 2020a. Incorporating Scenario Knowledge into A Unified Fine-tuning Architecture for Event Representation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pages 249–258, New York, NY, USA. Association for Computing Machinery.

Jianming Zheng, Fei Cai, Yanxiang Ling, and Honghui Chen. 2020b. Heterogeneous Graph Neural Networks to Predict What Happen Next. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 328–338, Barcelona, Spain (Online). International Committee on Computational Linguistics.