Multistructure Contrastive Learning for Pretraining Event Representation

Jianming Zheng, Fei Cai, Jun Liu, *Senior Member, IEEE*, Yanxiang Ling, and Honghui Chen

***Abstract*— Event representation aims to transform individual events from a narrative event chain into a set of low-dimensional vectors to help support a series of downstream applications, e.g., similarity differentiation and missing event prediction. Traditional event representation models tend to focus on single modeling perspectives and thus are incapable of capturing physically disconnected yet semantically connected event segments. We, therefore, propose a heterogeneous event graph model (HeterEvent) to explicitly represent such event segments. Furthermore, another challenge in traditional event representation models is inherited from the datasets themselves. Data sparsity and insufficient labeled data are commonly encountered in event chains, easily leading to overfitting and undertraining. Therefore, we extend HeterEvent with a multistructure contrastive learning framework (MulCL) to alleviate the training risks from two structural perspectives. From the sequential perspective, a sequential-view contrastive learning component (SeqCL) is designed to facilitate the acquisition of sequential characteristics. From the graph perspective, a graph-view contrastive learning component (GraCL) is proposed to enhance the robustness of graph training by comparing different corrupted graphs. Experimental results confirm that our proposed MulCL$_{[W+E]}$ model outperforms state-of-the-art baselines. Specifically, compared with the previously proposed supervised model HeterEvent$_{[W+E]}$ [Zheng *et al.* (2020)], MulCL$_{[W+E]}$ shows an average improvement of 5.3% in terms of accuracy for the inference-ability-based tasks. For the representation-ability-based tasks, MulCL$_{[W+E]}$ achieves an average improvement of 2.7% in terms of accuracy for the hard similarity tasks and an improvement of 4.1% in terms of the Spearman's correlation for the transitive sentence similarity task, respectively.**

***Index Terms*— Contrastive learning, event representation, graph pretraining, sequence pretraining.**

I. Introduction

IN DAILY life, human activities exhibit continuous streams of typical patterns and strict orders of progression. For instance, in the human activity of "*dining in a restaurant*" shown in Fig. 1, the execution of the successive event "*X eat food*" depends on the previous action "*X order food*". In early work [2], social science and psychology were mainly adopted to define such structural knowledge in the form of *scripts* or *Fillmorean frames*. However, limited by the need for manual acquisition and the complexity of the activity modeling, *script learning* developed slowly. Since the protagonist assumption was proposed [3], [4], the typical form used to define the structural knowledge has been further refined to a *narrative event chain*, i.e., a sequence of partially ordered events with a common protagonist that the events evolve around. Based on this assumption, the consideration of statistical concurrence relations can enhance the automatic extraction of narrative event chains from raw text.

Upon the definition of *narrative event chains*, the majority of approaches focus on *event representation* that transforms an extracted event chain into a series of low-dimensional vectors. However, the existing models cannot capture semantically connected yet physically disconnected event segments [1]. For example, in Fig. 1, three individual events "*X chased by Y*", "*X blocked by door*" and "*X arrested by Y*" are disconnected in their physical positions yet together constitute the arrest scenario. Therefore, we propose a heterogeneous-event graph neural network model (HeterEvent) [1] to capture such physically disconnected event segments. By means of node-to-node paths through the constructed heterogeneous graph, HeterEvent can explicitly represent the information interactions among homogeneous or heterogeneous nodes to model physically disconnected event segments.

Another challenge in traditional event representation models is the inability to overcome the training dilemma stemming from the datasets themselves. Since event chains are automatically extracted based on statistical concurrence, the lack of sufficient descriptive words and occurrences of duplicated words are widely encountered in the datasets generated in this way, which can easily lead to highly sparse event chains or event graphs. Limited to the sparse data on the training process, existing event representation models have to bear the overfitting risk, hardly generalizing to unseen event chains in the testing phase. Furthermore, the supervised training of traditional event representation models require abundant labeled data to guarantee training convergence, especially for the graph-based models, e.g., HeterEvent [1]. Unfortunately, unaffordable annotation cost for labeled data hinders the development of such supervised models for event representation. Therefore, we would like to develop an event representation model that can surmount abovementioned barriers without

Manuscript received September 29, 2021; revised May 4, 2022; accepted May 20, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61702526 and in part by the Postgraduate Scientific Research Innovation Project of Hunan Province under Grant CX20190034. An earlier version of this paper was presented at the Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020) [1] [DOI: 10.18653/v1/2020.coling-main.29]. *(Corresponding author: Fei Cai.)*

Jianming Zheng, Fei Cai, Yanxiang Ling, and Honghui Chen are with the Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China (e-mail: zhengjianming12@nudt.edu.cn; caifei08@nudt.edu.cn; lingyanxiang@nudt.edu.cn; chenhonghui@nudt.edu.cn).

Jun Liu is with the Department of Computer Science, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: liukeen@xjtu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3177641.

Digital Object Identifier 10.1109/TNNLS.2022.3177641

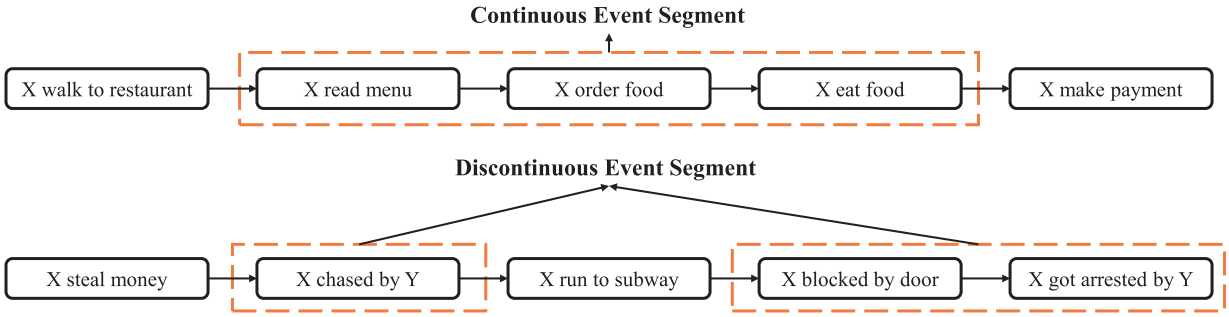


Fig. 1. Example of continuous and discontinuous event segments in two narrative event chains, where X and Y denote subject and object in a given event.

introducing external knowledge or requiring additional data input. Contrastive self-supervised learning [5], which leverages the differences among the input data themselves for supervision to achieve enhanced performance in representation learning and intuitively provides a solution for event representation.

Revisiting the HeterEvent model, we find that it has two different characteristic structures, i.e., the sequential structure of the initial event chain and the graph structure of the constructed heterogeneous graph. Hence, we attempt to address the issues of data sparsity and undertraining in HeterEvent by proposing a **Mul**tistructure **C**ontrastive **L**earning framework (MulCL) for pretraining on the original event chains and the constructed heterogeneous graphs. In particular, from the sequential perspective, we design a **Seq**uential-view **C**ontrastive **L**earning component (SeqCL), which focuses on the structure and content contrasts of the initial event chains. SeqCL employs three kinds of contrast strategies, considering order-based, replacement-based, and composition-based contrasts, to compare intra-event elements and individual events, which can help capture certain sequential characteristics. From the graph-structure perspective, we propose a **Gra**ph-view **C**ontrastive **L**earning component (GraCL) to enhance the robustness of graph training. Specifically, GraCL adopts some graph augmentation methods [6] to generate a series of corrupted HeterEvent graphs. By mutually comparing the normal and corrupted graphs or different corrupted graphs, GraCL can mitigate the risk of inadequate training due to the presence of limited supervised signals in the sparse HeterEvent graphs. Furthermore, MulCL combines SeqCL and GraCL in a sequential training process, which enables the HeterEvent model to capture multistructure characteristics in the pretraining stage.

Following Zheng *et al.* [1], we evaluate the inference ability of our proposed model for the one-step [7] and multistep [8] inference tasks. After that, we present a deep assessment of our proposal in terms of its representation ability by focusing on the hard similarity [9] and transitive sentence similarity [10] tasks. The experimental results demonstrate that MulCL can improve the representation and inference capabilities upon the purely supervised event model HeterEvent$_{[W+E]}$ [1]. In addition, an ablation study verifies the effectiveness of each pretraining component. In particular, SeqCL is better at enhancing the representation ability while GraCL excels at improving the inference ability.

In summary, our research offers the following three contributions.

1) Without the extra data input, we propose an SeqCL to facilitate the acquisition of sequential characteristics, including the event-chain order, the event-chain composition, and local–global event information.
2) We propose a GraCL that compares different corrupted HeterEvent graphs to compensate for the lack of sufficient supervised signals in the training process.
3) In addition to the inference-ability tasks for evaluation, we also examine our proposals for three kinds of representation-ability tasks. Compared with HeterEvent$_{[W+E]}$ [1], the introduction of MulCL generally leads to an improvement of 2.2%–7.2% in terms of accuracy or the Spearman's correlation for the inference- and representation-ability-based tasks.

II. Related Work

A. Narrative Event Chains

Event chains model human understanding of the relevant causal relationships among events. An event chain can be used to infer how events will unfold in a given scenario [2]. Restricted to manual acquisition, early work on event chains showed slow progress until narrative event chains [3] were introduced, which assumes that although a narrative script typically has several participants, there is one central actor (a.k.a protagonist) who characterizes a narrative chain. Consequently, probabilistic co-occurrence-based models combined with a dependency parser can realize the automatic extraction of narrative event chains from raw text [3], representing the narrative events in the format of $< predicate, dependency\_type >$, where the *predicate* is a verb lemma and the *dependency_type* specifies a grammatical dependency relation between the *predicate* and the protagonist, e.g., "subj," "obj," or "iobj." In addition, a richer representation over a multiargument event format has been explored [11].

According to the modeling perspective, existing work on event representation can be mainly classified into three types, i.e., *intra-event*-based, *inter-event*-based and *event-segment*-based models. First, intra-event-based methods focus on multiplicative interactions among intra-event elements. For instance, Granroth-Wilding and Clark [7] simply concatenated on predicate and argument embeddings and fed them into a neural network to obtain the event representation, whereas Weber *et al.* [9] used a tensor-network-based model to capture subtle semantic interactions. Second, inter-event-based methods mainly investigate complex and diverse relations

between two individual events. For instance, long short-term memory (LSTM) hidden states can be utilized to integrate chain-order information into an event model [12]. Narrative event chains can also be extended to narrative event evolutionary graphs to model the dense connections among events [13]. In addition, Lee and Goldwasser [14] broadened relations of a single type (time-order relations) into diverse relations based on discourse relations from the Penn Discourse Treebank (PDTB) [15]. Finally, event-segment-based methods focus on sets of semantically related events. Lv *et al.* [16] included a self-attention mechanism developed to implicitly model relations in event segments. Zheng *et al.* [17] employed a unified fine-tuning framework to integrate the training losses from different layers. In addition, by jointly training an event representation model with external knowledge, some work has attempted to mine potential connections between narrative event chains and external knowledge [8], [18]. For example, Lee and Goldwasser [8] explored ATOMIC [18] to obtain sentiment and intent information in events.

Different from the above methods, our work attempts to synthetically represent multigranularity information and discontinuous event segments contained in an event chain by means of a heterogeneous graph network, which can provide strong inference capabilities for event prediction. It is also worth noting that the HeterEvent graph is a scalable framework that can be easily adjusted to enable the fusion of information at additional levels of granularity, e.g., subwords or event scenarios.

B. Contrastive Learning

Due to the high cost of annotation, insufficient availability of labeled data is widely encountered in reality. However, contrastive learning can readily overcome this dilemma through comparisons of unlabeled data. For instance, recent breakthroughs in computer vision (e.g., Deep InfoMax [19], MoCo [20], and SimCLR [21]) have pushed the performance of unsupervised image classification to approach and even exceed that of supervised methods. The core of contrastive learning is to pull neighbors closer together and push non-neighbors farther apart [22]. Oord *et al.* [23] concentrated on contrastive predictive coding and transformed the intractable computation of mutual information into a simpler softmax format, whereas contrastive multiview coding [24] extends such a contrastive learning framework to the multiview setting, maximizing the mutual information between different views for the same target object. Furthermore, Deep Graph InfoMax [25] adapts contrastive learning to a general graph domain, leveraging local mutual information maximization across different graph patch representations.

In the context of natural language processing, the application of contrastive learning dates back to noise-contrastive estimation [26], in which negative sampling is employed to obtain word embeddings, dramatically reducing the computational cost. On this basis, the skip-gram approach [27] follows the distributed assumption that a good representation of a word should be predictive of its context to form semantically abundant word embeddings (i.e., word2vec). BERT [28] relies on two contrast-based subtasks, i.e., masked language modeling and next sentence prediction, to realize encoder training. InfoWord [29] adopts a "global–local" comparison approach, in which the context of words (i.e., the [CLS] output of BERT) and the words themselves are treated as global and local information, respectively. CONPONO [30] considers discourse coherence and constructs objects for comparison based on the distances between sentences. Furthermore, the objects for comparison in contrastive learning are usually selected as the same target under different augmentations; however, the discrete nature of natural language exacerbates the difficulty of creating a label-preserving transformation for text data. Hence, CoDA [31] was proposed as a means of performing synthesized data augmentation by organically integrating multiple transformations.

The aforementioned models mainly focus on single structural characteristics of the objects for comparison, neglecting other structural information. As an extension of these methods, this article considers not only the comparison of the sequential structures of the initial event chains but also the graph structures constructed using HeterEvent. Integrating multistructural characteristics for contrastive pretraining can overcome the challenges of sparsity and undertraining faced during the training of the HeterEvent model.

III. METHODOLOGY

We present the pretraining and training processes of the HeterEvent graph model in Fig. 2, which can be divided into steps performed by the following five components: an encoding layer, a graph layer, a SeqCL, a graph-view augmentation component, and a prediction layer.

A. Problem Definition

Given a set of training event chains $\mathbb{T}$, a *narrative event chain* $\mathcal{T}$ in $\mathbb{T}$ models a typical human activity, usually consisting of a series of sequentially related individual events e (i.e., $\mathcal{T} = \{e_i\}_{i=1}^{n}$). In the event chain $\mathcal{T}$, each individual event e describes a specific human action, which can be represented by an intra-event triplet consisting of a predicate p, a subject s and a object o, i.e., $e = \{p, s, o\}$.

Through graph modeling, each sequential event chain can be transformed into a heterogeneous graph structure, i.e., the HeterEvent graph. Let a HeterEvent graph be denoted by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of edges between nodes. In particular, we treat each individual word or event as a node (i.e., $\mathcal{V} = \mathcal{V}_w \cup \mathcal{V}_e$) and define three types of undirected edges between pairs of nodes to model various kinds of structural information in the HeterEvent graph (i.e., $\mathcal{E} = \{\mathcal{E}_{w-w} \cup \mathcal{E}_{w-e} \cup \mathcal{E}_{e-e}\}$). Here, $\mathcal{V}_w = \{w_1, w_2, \ldots, w_m\}$ denotes the m unique words in an event chain, and $\mathcal{V}_e = \{e_1, e_2, \ldots, e_n\}$ corresponds to n individual events in the event chain.

In this article, we aim to learn an event representation to investigate its representation and inference capabilities. On the one hand, given an incomplete event chain $\mathcal{T}' = \{e_1, \ldots, e_{i-1}, e_{i+1}, \ldots, e_n\}$ and a set of candidate events $\{e_i^c\}_{i=1}^{m}$, the inference ability refers to the ability to choose the correct event from among the candidate events for the missing event e_i in $\mathcal{T}'$. On the other hand, the representation

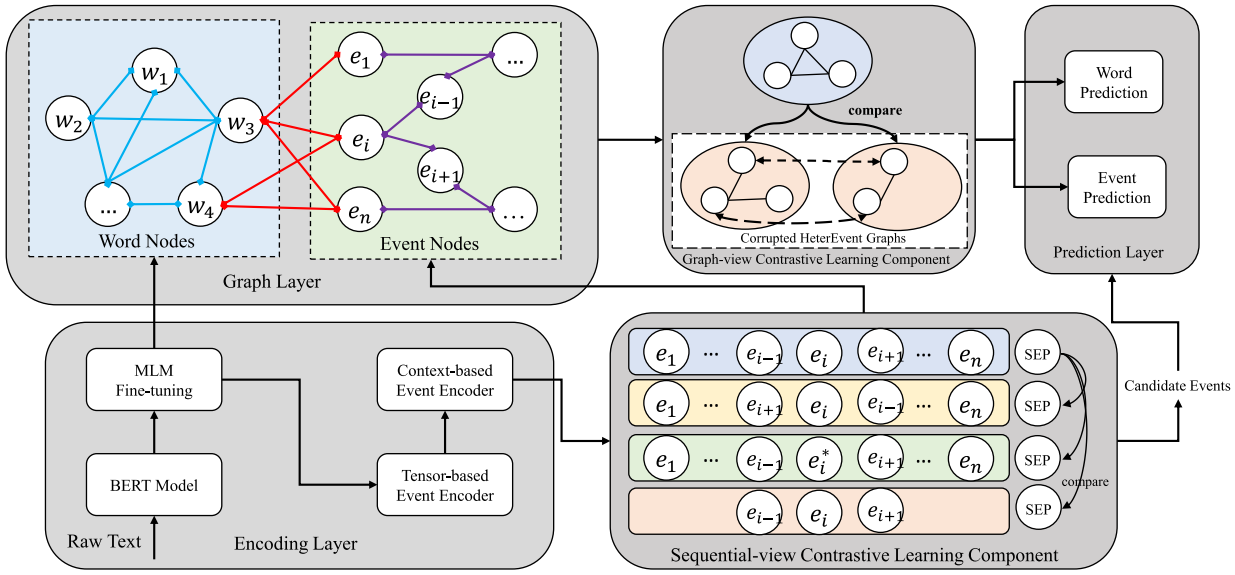


Fig. 2. Pretraining and training processes in the HeterEvent graph model. In SeqCL, the blue rectangle represents the normal event chain, while the yellow, green, and orange rectangles denote the order-corrupted, replacement-corrupted, and composition-corrupted event chains, respectively. In the graph layer, the blue, red, and purple lines represent the word–word, word–event, and event–event edges, respectively. In GraCL, the blue and orange ellipses represent the normal and corrupted HeterEvent graphs, respectively.

ability refers to the expressiveness of the encoder. Given a similarity function $\mathcal{S}$, any similar event pair $\{e_i^+, e_j^+\}$ and any dissimilar event pair $\{e_i^-, e_j^-\}$, the event encoder $\mathcal{F}$ should satisfy $\mathcal{S}(\mathcal{F}(e_i^+), \mathcal{F}(e_j^+)) \gg \mathcal{S}(\mathcal{F}(e_i^-), \mathcal{F}(e_j^-))$.

B. Multistructure Contrastive Learning Framework

This section first introduces the encoding process for the initial event chain and the HeterEvent graph in Section III-B1. The pretraining of the initial embeddings by SeqCL and GraCL is described in Sections III-B2 and III-B3, respectively.

1) Context-Based Event Encoder: To overcome the inconsistency of the pretraining corpora (e.g., the BERT model was pretrained on BooksCorpus [32] and English Wikipedia [28], whereas the narrative event chains were pretrained on the Gigaword corpus [33]), the context-based event encoder first employs a masked language model [28] to minimize the influence of this corpus inconsistency. Similar to Devlin *et al.* [28], we randomly mask some words in a text sequence with the $[MASK]$ token and feed them into the BERT model to predict the masked words. In comparison with previous methods, e.g., [7], [9], [16], in which GloVe [34] is directly applied to word representation, such fine-tuning narrows the semantic distribution gap when the model is transferred to different corpora.

With the fine-tuned BERT model, we can then directly obtain the representations of the intra-event elements. Since each intra-event element is likely to contain multiple words, we apply the max pooling and average pooling to those words and then concatenate the results to obtain the representations for these three types of intra-event elements (i.e., the subject, predicate, and object representations are denoted by $s(e), p(e), o(e) \in \mathbb{R}^{2d}$, respectively). Specifically, the subject representation $s(e)$ is defined as follows:

$$s(e) = \left[\max\left(\left[w_{s,1}, \ldots, w_{s,n_s}\right]\right), \text{ave}\left(\left[w_{s,1}, \ldots, w_{s,n_s}\right]\right)\right]$$

where $w_{s,1}, w_{s,2}, \ldots, w_{s,n_s} \in \mathbb{R}^d$ are the representations of the words in the subject, and $\max(\cdot)$, $\text{ave}(\cdot)$, and $[;]$ denote the max pooling, average pooling and concatenation operations, respectively. The same strategy is also applied to obtain the representations of the predicate $p(e)$ and the object $o(e)$. In our pilot experiments, such a hybrid pooling process is more effective than the traditional average pooling and the max-pooling.

Following Weber *et al.* [9], we adopt a tensor-based model [35] to explore the subtle semantic interactions among intra-event elements. Given a 3-D tensor-based network $T(,)$ with inputs a and b, where $T \in \mathbb{R}^{d\times 2d\times 2d}$ and $a, b \in \mathbb{R}^{2d}$, we compute the result $T(a,b) = \sum_{j,k} T_{i,j,k} a_j b_k$. Hence, the representation $E(e)$ for each individual event is formulated as

$$E(e) = W_s T(s(e), p(e)) + W_o T(o(e), p(e)) \tag{1}$$

where $W_s, W_o \in \mathbb{R}^{d\times d}$ are the trade-off matrices for the subject role and the object role, respectively.

After that, we use a bidirectional gated recurrent unit (Bi-GRU) on top of the event encoder to model the temporal interactions between events, i.e., forward- and backward-order information [12]. Then, we can obtain a sequence of hidden-state representations $\{h_1, h_2, \ldots, h_n\}$ by recurrently feeding the event representations $\{E(e_1), E(e_2), \ldots, E(e_n)\}$ as inputs into Bi-GRU, i.e.,

$$\begin{cases} \overrightarrow{h_i} = \overrightarrow{\text{GRU}}\left(E(e_i), \overrightarrow{h_{i-1}}\right) \\ \overleftarrow{h_i} = \overleftarrow{\text{GRU}}\left(E(e_i), \overleftarrow{h_{i-1}}\right) \end{cases} \tag{2}$$

where $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}]$; h_0 and other parameters in Bi-GRU are randomly initialized.

2) Sequential-View Contrastive Learning Component: When attempting to enrich the event representations with sequential characteristics, relying solely on a Bi-GRU may

lead to two problems. First, a traditional Bi-GRU cannot avoid the risk of information loss as the length of the event chain increases. Second, a sparse event chain with duplicated intra-event elements can easily cause the event representation model to become trapped in a local optimum, resulting in undertraining of the model. Hence, as shown in *SeqCL* in Fig. 2, we define three aspects of sequential contrast, namely, order contrast, replacement contrast and composition contrast, and overcome the aforementioned problems by comparing the normal event chain with three corresponding kinds of corrupted chains.

a) Order contrast: In narrative event chains, strict event-order relations typically hold, influencing what happens next. For example, a chronological-order relation determine that the event "*X order food*" cannot occur after the event "*X eat food*." Therefore, we design an order contrast mechanism to capture not only such order relations between adjacent events but also the overall development-order relation of the whole event chain.

For any given event chain $\mathcal{T} = \{e_i\}_{i=1}^{n}$, we first randomly change the partial order of events (e.g., $\{e_1, \ldots, e_{i+1}, e_i, e_{i-1}, \ldots, e_n\}$) to produce a set of N_c order-corrupted event chains $\mathcal{N}_{\text{order}}^{\mathcal{T}} = \{\mathcal{T}_i^O\}_{i=1}^{N_c}$, where $\mathcal{T}_i^O$ denotes the ith order-corrupted event chain and the exchange probability for all events follows the Bernoulli distribution $\mathcal{P}_B(n, 0.2)$. To obtain the global information, we append a special token [SEP] to each event chain. Then, we feed the lengthened event chain into the context-based event encoder and treat the output for the token [SEP] as the global representation of the event chain, i.e., h_S. In this way, for the initial event chain and the set of order-corrupted event chains, we obtain the corresponding global representations $h_S^{\mathcal{T}}$ and $\{h_S^{\mathcal{T}_i^O}\}_{i=1}^{N_c}$. Similar to the InfoNCE loss [23], we calculate the order-contrast loss $\mathcal{L}_{\text{order}}^{\mathcal{S}}$ as

$$\mathcal{L}_{\text{order}}^{\mathcal{S}} = -\frac{1}{|\mathbb{T}|}\sum_{\mathcal{T}\in\mathbb{T}}\frac{e^{\mathcal{S}\left(h_S^{\mathcal{T}}, h_S^{\mathcal{T}}\right)/\tau}}{\sum_{i=1}^{N_c} e^{\mathcal{S}\left(h_S^{\mathcal{T}}, h_S^{\mathcal{T}_i^O}\right)/\tau} + e^{\mathcal{S}\left(h_S^{\mathcal{T}}, h_S^{\mathcal{T}}\right)/\tau)}} \quad (3)$$

where $\mathcal{S}$ denotes the cosine similarity and τ represents the softmax temperature.

b) Replacement contrast: In a multistep inference task, e.g., MCNS-V or MCNE-V [8], different start events trigger different event chains. This finding leads to a more general hypothesis that any individual event can determine the development of the event chain. Hence, the concept of replacement contrast is defined to capture any anomalous event in the event chain.

Given an event chain $\mathcal{T} = \{e_i\}_{i=1}^{n}$, we first randomly select some events to be replaced based on the Bernoulli distribution $\mathcal{P}_B(n, 0.2)$ and substitute them with other unrelated events from other event chains. For instance, the replacement-corrupted event chain $(\{e_1, \ldots, e_{i-1}, e_i^*, e_{i+1}, \ldots, e_n\})$ in Fig. 2 is formed by replacing e_i with e_i^*. In this way, we obtain a set of N_c replacement-corrupted event chains $\mathcal{N}_{\text{replace}}^{\mathcal{T}} = \{\mathcal{T}_i^R\}_{i=1}^{N_c}$, where $\mathcal{T}_i^R$ denotes the ith replacement-corrupted event chain. Then, we similarly append a special token [SEP] to the end of each event chain and feed these lengthened event chains into the context-based event encoder. The output of the token [SEP] can be regarded as the global representation of the event chain, i.e., $h_S^{\mathcal{T}}$ and $\{h_S^{\mathcal{T}_i^R}\}_{i=1}^{N_c}$ for the unprocessed event chain and the set of replacement-corrupted event chains, respectively. With the global representations, we can obtain the replacement-contrast loss $\mathcal{L}_{\text{replace}}^{\mathcal{S}}$ as

$$\mathcal{L}_{\text{replace}}^{\mathcal{S}} = -\frac{1}{|\mathbb{T}|}\sum_{\mathcal{T}\in\mathbb{T}}\frac{e^{\mathcal{S}\left(h_S^{\mathcal{T}}, h_S^{\mathcal{T}}\right)/\tau}}{\sum_{i=1}^{N_c} e^{\mathcal{S}\left(h_S^{\mathcal{T}}, h_S^{\mathcal{T}_i^R}\right)/\tau} + e^{\mathcal{S}\left(h_S^{\mathcal{T}}, h_S^{\mathcal{T}}\right)/\tau}}. \quad (4)$$

c) Composition contrast: Work on Deep InfoMax [19] has revealed that maximizing the mutual information between the local and global feature maps can improve the representation. In addition, work on SAM-Net [16] has proven that not every event contributes equally to the prediction of the next event; instead, some partial event segments matter more. Inspired from these findings, the composition contrast mechanism is designed to remove noisy events and locate important event segments.

For any given event chain $\mathcal{T} = \{e_i\}_{i=1}^{n}$, we can randomly extract several continuous events from the whole event chain (e.g., $\mathcal{P}_{\text{seg}}^i = \{e_{i-1}, e_i, e_{i+1}\}$ in Fig. 2) to produce a positive set of segments $\mathbb{P}_{\text{seg}}^{\mathcal{T}} = \{\mathcal{P}_{\text{seg}}^i\}_{i=1}^{N_s}$. For each event segment $\mathcal{P}_{\text{seg}}^i$, we can then change the order of events or replace an event to produce composition-corrupted event chains, i.e., $\mathcal{N}_{\text{compo}}^{\text{seg}_i,\mathcal{T}} = \{\mathcal{T}_{\text{seg}_i,j}^C\}_{j=1}^{N_c}$, where $\mathcal{T}_{\text{seg}_i,j}^C$ denotes the jth composition-corrupted event chain extracted from event segment $\mathcal{P}_{\text{seg}}^i$. We then append the special token [SEP] to the end of all event chains, including the initial event chain $\mathcal{T}$, the event segments $\mathbb{P}_{\text{seg}}^{\mathcal{T}}$ and the composition-corrupted event chains $\{\mathcal{N}_{\text{compo}}^{\text{seg}_i,\mathcal{T}}\}_{i=1}^{N_s}$. Correspondingly, the representations from the output token [SEP] can be obtained as $h_S^{\mathcal{T}}$, $\{h_S^{\mathcal{P}_{\text{seg}}^i}\}_{i=1}^{N_s}$, and $\{\{h_S^{\mathcal{T}_{\text{seg}_i,j}^C}\}_{j=1}^{N_c}\}_{i=1}^{N_s}$, respectively. Unlike that in the order contrast and replacement contrast mechanisms, we adopt $\mathcal{T}$ and $\mathbb{P}_{\text{seg}}^{\mathcal{T}}$ to construct the positive pairs, while $\mathcal{T}$ and $\{\mathcal{N}_{\text{compo}}^{\text{seg}_i,\mathcal{T}}\}_{i=1}^{N_s}$ form the negative pairs. Then, the composition-contrast loss $\mathcal{L}_{\text{comp}}^{\mathcal{S}}$ is formulated as

$$\mathcal{L}_{\text{comp}}^{\mathcal{S}} = -\frac{1}{|\mathbb{T}|}\sum_{\mathcal{T}\in\mathbb{T}}\sum_{\mathcal{P}_{\text{seg}}^i\in\mathbb{P}_{\text{seg}}^{\mathcal{T}}}\frac{e^{\mathcal{S}\left(h_S^{\mathcal{T}}, h_S^{\mathcal{P}_{\text{seg}}^i}\right)/\tau}}{\sum_{j=1}^{N_c} e^{\mathcal{S}\left(h_S^{\mathcal{T}}, h_S^{\mathcal{T}_{\text{seg}_i,j}^C}\right)/\tau} + e^{\mathcal{S}\left(h_S^{\mathcal{T}}, h_S^{\mathcal{P}_{\text{seg}}^i}\right)/\tau}}. \quad (5)$$

Finally, the overall loss for SeqCL can be constructed as a joint training loss, i.e.,

$$\mathcal{L}^{\mathcal{S}} = \lambda_1\mathcal{L}_{\text{order}}^{\mathcal{S}} + \lambda_2\mathcal{L}_{\text{replace}}^{\mathcal{S}} + \lambda_3\mathcal{L}_{\text{comp}}^{\mathcal{S}} \quad (6)$$

where λ_1–λ_3 are the trade-off parameters such that $\sum_{i=1}^{3}\lambda_i = 1$.

We detail the process of SeqCL in Algorithm 1, where three types of corrupted event chains are constructed in lines 3–8. Specifically, the training process in line 8 reveals that this contrastive learning procedure does not introduce additional parameters into the HeterEvent training process and affects only the encoder layer. With the self-supervised signals obtained

Algorithm 1 SeqCL

Input: A training set of event chains $\mathbb{T}$; the parameter set Θ for the context-based event encoder.
Output: Fine-tuned Θ.

```
for each event chain 𝒯 ∈ 𝕋 do
    Embed 𝒯 using the context-based event encoder;
    Randomly change the event order in 𝒯: 𝒩^𝒯_order;
    Randomly replace events in 𝒯: 𝒩^𝒯_replace;
    Extract continuous events from 𝒯: ℙ^𝒯_seg = {𝒫^i_seg}_{i=1}^{N_s};
    for i=1; i ≤ N_s; i++ do
        Create composition-corrupted event chains for 𝒫^i_seg: 𝒩^{seg_i,𝒯}_compo;
    end for
    Update Θ based on Eq.(6);
end for
Return the fine-tuned Θ.
```

from these three contrast mechanisms, the undertraining risk stemming from sparse data can be effectively mitigated.

3) Graph-View Contrastive Learning Component: Based on the pretrained sequential-view event encoder, we can construct a heterogeneous graph with two kinds of nodes and three kinds of edges. The next step is to allow the information of each node to pass to other nodes via edges.

Formally, given a node i and its set of neighborhood nodes $\mathcal{N}_i$, the output of neighborhood aggregation for node i in layer k can be formulated as follows:

$$z_i^k = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij} h_j^k\right) \tag{7}$$

where σ denotes the sigmoid function and h_j^k is the node representation of node j in layer k. Similar to Velickovic *et al.* [36], α_{ij} is the attention weight between h_i^k and h_j^k, which is defined as

$$\alpha_{ij} = \frac{e^{\text{LeakyReLU}\left(W_a\left[W_s h_i^k; W_s h_j^k\right]\right)}}{\sum_{l\in\mathcal{N}_i} e^{\text{LeakyReLU}\left(W_a\left[W_s h_i^k; W_s h_l^k\right]\right)}} \tag{8}$$

where LeakyReLU$(\cdot)$ is a nonlinear function; W_a and W_s are trainable weight matrices, and $[;]$ is the concatenation operation. For graph neural networks, when the number of layers is too large, i.e., neighborhood aggregation is performed too many times, the oversmoothing problem [37] can easily arise. Hence, we add the residual connections [38] to alleviate this issue

$$u_i^k = z_i^k + h_i^k. \tag{9}$$

In addition, we introduce an information gate g_i^k to control the process of updating the node representation h_i^k [39]. Accordingly, the updated representation of node i, i.e., the representation of node i in layer $k+1$, can be expressed as

$$\begin{cases} h_i^{k+1} = g_i^k \odot \tanh\left(u_i^k\right) + \left(1 - g_i^k\right) \odot h_i^k \\ g_i^k = \sigma\left(W_g\left[u_i^k; h_i^k\right]\right) \end{cases} \tag{10}$$

where $\odot$ denotes an elementwise multiplication, σ is the sigmoid function, and W_g is a trainable weight matrix. Hence, following such an information passing strategy, adding one information-passing layer can enable information aggregation for one-hop neighborhood nodes, i.e., continuous nodes. That is, the addition of one more information-passing layer is equivalent to implementing information aggregation for one-more-hop neighborhood nodes, which helps information pass to discontinuous nodes.

However, direct training of such a HeterEvent graph suffers from the limited availability of labeled data, making model convergence difficult to realize. Therefore, we consider two aspects of graph contrast, namely, normal–corrupted contrast and corrupted–corrupted contrast, to provide rich self-supervised signals.

a) Normal–corrupted contrast: Given a HeterEvent graph $\mathcal{G}$, we adopt the augmentation operation $\mathcal{A}$ [6] for graph contrastive learning to produce a set of corrupted HeterEvent graphs $\mathcal{A}(\mathcal{G}) \longmapsto \mathbb{G}$ ($\mathbb{G} = \{\mathcal{G}_i'\}_{i=1}^{N_g}$), where $\mathcal{G}_i'$ denotes the ith corrupted graph with nodes and edges represented by $\mathcal{V}_i'$ and $\mathcal{E}_i'$, respectively. Then, we feed these normal and corrupted HeterEvent graphs into the message-passing layer to realize node interaction, obtaining the corresponding node representations. In addition, we employ a simple averaging of all nodes' representations as the readout function for the normal HeterEvent graph [19], i.e.,

$$g_{\text{read}} = \sigma\left(\sum_{i=1}^{|\mathcal{V}|} h_i\right) \tag{11}$$

where $\{h_i\}_{i=1}^{|\mathcal{V}|}$ denotes the representations of all nodes in $\mathcal{V}$ after the message passing process.

Similar to the Graph InfoMax approach [19], we employ the readout representation g_{read} and the node representations $\{h_i\}_{i=1}^{|\mathcal{V}|}$ from the normal HeterEvent graph as positive pairs, while the readout representation g_{read} and the node representations $\{h'_{i,j}\}_{j=1}^{|\mathcal{V}_i'|}(i = 1, \ldots, N_g)$ from the corrupted HeterEvent graphs as negative pairs. Accordingly, the normal–corrupted contrast loss $\mathcal{L}_{\text{Nor}}^{\mathcal{G}}$ is formulated as follows:

$$\mathcal{L}_{\text{Nor}}^{\mathcal{G}} = -\frac{1}{|\mathbb{T}|}\sum_{\mathcal{G}\in\mathbb{T}} \frac{\sum_{i=1}^{|\mathcal{V}|} e^{\mathcal{S}(g_{\text{read}}, h_i)/\tau}}{\sum_{i=1}^{N_g}\sum_{j=1}^{|\mathcal{V}_i'|} e^{\mathcal{S}\left(g_{\text{read}}, h'_{i,j}\right)/\tau} + \sum_{i=1}^{|\mathcal{V}|} e^{\mathcal{S}(g_{\text{read}}, h_i)/\tau}} \tag{12}$$

where the definitions of $\mathcal{S}$ and τ are shown in (3)–(5). However, such contrastive learning driven only by the readout representation g_{read} is likely to overly smooth the positive node representations and reduce the node diversity. Therefore, we introduce a corrupted–corrupted contrast calculation to increase the node diversity.

b) Corrupted–corrupted contrast: The goal of the corrupted–corrupted contrast calculation is to distinguish the same node in different corrupted HeterEvent graphs and prevent the representations of different nodes from collapsing into similar ones. Therefore, we can treat the instances of the same node in different corrupted graphs as a positive pair and

Algorithm 2 GraCL

Input: A training set of event chains $\mathbb{T}$; the parameter set Φ for the message-passing layer in the HeterEvent graph; the parameter set Θ for the fine-tuned context-based encoder.
Output: Fine-tuned Φ and Θ.
1: **for** each event chain $\mathcal{T} \in \mathbb{T}$ **do**
2: Construct the HeterEvent graph $\mathcal{G}$ for $\mathcal{T}$;
3: Construct a set of corrupted graphs $\mathbb{G}$;
4: **for** $i = 1; i \leq N_g; i++$ **do**
5: Compute the normal–corrupted contrast between $\mathcal{G}'_i$ and $\mathcal{G}$ based on Eq.(12);
6: **for** $j = i + 1; j \leq N_g; j++$ **do**
7: Compute the corrupted–corrupted contrast between $\mathcal{G}'_i$ and $\mathcal{G}'_j$ based on Eq.(12);
8: **end for**
9: **end for**
10: Update Φ and Θ based on Eq.(14);
11: **end for**
12: Return the fine-tuned Φ and Θ.

different nodes from either the same or different corrupted graphs as a negative pair.

In this way, the corrupted–corrupted contrast loss can be formulated as follows:

$$\mathcal{L}_{\text{Cor}}^{\mathcal{G}} = -\frac{1}{|\mathbb{T}|}\sum_{\mathcal{G}\in\mathbb{T}}\sum_{\mathcal{V}'_i,\mathcal{V}'_j\in\mathbb{G}}\sum_{k=1}^{N_v} \times \frac{e^{\mathcal{S}\left(h'_{i,k},h'_{j,k}\right)/\tau}}{e^{\mathcal{S}\left(h'_{i,k},h'_{j,k}\right)/\tau} + \sum_{l=1}^{N_v} e^{\mathcal{S}\left(h'_{i,k},h'_{i,l}\right)/\tau} + e^{\mathcal{S}\left(h'_{i,k},h'_{j,l}\right)/\tau}} \tag{13}$$

where N_v denotes the number of nodes shared between $\mathcal{V}'_i$ and $\mathcal{V}'_j$ under two corrupted HeterEvent graphs $\mathcal{G}'_i$ and $\mathcal{G}'_j$, while $e^{\mathcal{S}(h'_{i,k},h'_{j,k})/\tau}$ denotes the similarity score for the kth shared node between these two corrupted graphs. In addition, $e^{\mathcal{S}(h'_{i,k},h'_{i,l})/\tau}$ and $e^{\mathcal{S}(h'_{i,k},h'_{j,l})/\tau}$ represent the similarity scores between different nodes from the same corrupted graph and between different nodes from different corrupted graphs, respectively.

Finally, we again employ a joint training to combine these graph-view contrast losses, i.e.,

$$\mathcal{L}^{\mathcal{G}} = \alpha_1\mathcal{L}_{\text{Nor}}^{\mathcal{G}} + \alpha_2\mathcal{L}_{\text{Cor}}^{\mathcal{G}} \tag{14}$$

where α_1 and α_2 are the trade-off parameters such that $\alpha_1 + \alpha_2 = 1$.

We present the pseudocode of GraCL in Algorithm 2, in which two kinds of contrasts are computed from lines 4–9. This training process updates only the original parameters in the initial HeterEvent graph without any external knowledge input. Therefore, such contrastive self-supervised training can help the HeterEvent graph converge with limited labeled data.

C. HeterEvent Training

We have obtained the node representations of the events, i.e., $\{h_{e_1}, h_{e_2}, \ldots, h_{e_n}\}$. For h_{e_i} $(i = 1, 2, \ldots, n)$, the node representations of its subordinate words are $\{w_1^{e_i}, w_2^{e_i}, \ldots, w_{n_{e_i}}^{e_i}\}$, where n_{e_i} is the number of words in e_i. In the supervised training phase, we train our HeterEvent model at two levels: the word level and the event level.

At the word level, the node representations of words can be used to predict the neighborhood word nodes, corresponding to the training of $\mathcal{E}_{e-e}$. On the other hand, the word nodes can be inferred from the node representations of their source events, corresponding to the training of $\mathcal{E}_{w-e}$. Therefore, the loss function at the word level can be formulated as follows:

$$\mathcal{L}_w = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n_{e_i}} -\log\left(P\left(w_j^{e_i}\middle|w_1^{e_i}\right)P\left(w_j^{e_i}\middle|h_{e_i}\right)\right) + \lambda\mathcal{L}_w(\theta_w) \tag{15}$$

where $P(w_j^{e_i}|h_{e_i})$ and $P(w_j^{e_i}|h_{e_i})$ are computed via a softmax layer, λ is a trade-off parameter, and $\mathcal{L}_w(\theta_w)$ denotes $l2$ regularization on all parameters θ_w. At the event level, the node representation of event h_{e_i} can be used to predict the previous and next events, corresponding to the training of $\mathcal{E}_{s-s}$. Similarly, the loss function at the event level can be formulated as follows:

$$\mathcal{L}_e = \frac{1}{n}\sum_{i=1}^{n} -\log\left(P\left(h_{e_{i-1}}|h_{e_i}\right)P\left(h_{e_{i+1}}|h_{e_i}\right)\right) + \lambda\mathcal{L}_e(\theta_e) \tag{16}$$

where $P(h_{e_{i-1}}|h_{e_i})$ and $P(h_{e_{i+1}}|h_{e_i})$ are computed via a softmax layer—specifically, $P(h_{e_0}|h_{e_1}) = 1$—and $\mathcal{L}_w(\theta_e)$ denotes $l2$ regularization on all parameters θ_e. Intuitively, we can combine these two individual losses through simple addition, i.e., $\mathcal{L}_{w+e} = \mathcal{L}_w + \mathcal{L}_e$, to evaluate the model performance. Furthermore, we employ a noise-contrastive estimation [26] to reduce the computational cost.

In the testing phase, the whole event chain is passed through the encoder layer and the graph layer to construct the HeterEvent graph, while candidate events are passed only through the encoder layer to obtain the corresponding representations, which are then combined with the constructed graph to select the most likely event based on their similarity scores. In summary, Algorithm 3 presents the pseudocode of the supervised training process for the HeterEvent graph. In detail, the sequential and graph characteristics are initialized by the fine-tuned Θ and Φ from Algorithms 1 and 2, respectively, from lines 2–4. For each event chain, the event representation model is further trained under supervision from the ground-truth missing event in line 5.

IV. Experiments

A. Ability Evaluation

We evaluate the inference ability of our proposed models for two types of inference tasks: one-step and multistep inference tasks.

1) *One-Step Inference Task:* The aim is to predict a missing event given its context. On this basis, the multiple-choice narrative cloze (MCNC) dataset is proposed [7].
2) *Multistep Inference Task:* As an extension of the one-step inference task, this task is designed to evaluate a model's ability to make longer inferences instead of predicting only one event. Lee and Goldwasser [8] proposed three

Algorithm 3 Supervised Training for the HeterEvent Graph

Input: A training set of event chains $\mathbb{T}$ with missing events; the fine-tuned context-based event encoder parameters Θ from Algorithm 1; the fine-tuned message-passing layer parameters Φ from Algorithm 2.

Output: Trained Θ and Φ.

```
for each event chain 𝒯 ∈ 𝕋 do
    Embed 𝒯 using the context-based event encoder Θ;
    Construct the corresponding HeterEvent graph 𝒢;
    Realize the message-passing layer with the parameters Φ in 𝒢;
    Predict each missing event and its subordinate words and then update Θ and Φ based on Eq.(15) and Eq.(16);
end for
Return the trained Θ and Φ.
```

selection strategies for constructing event chains, i.e., *Viterbi*, *Base* and *Sky*. *Viterbi* considers the integrity of the event chain and finds the most likely event chain, *Base* greedily picks the best transition and then moves to the next time stamp, and *Sky* breaks down a sequence of predictions into individual decisions in which the golden states of all contextual events are applied. Hence, these three selection strategies can be used to build four types of multi-inference datasets, i.e., MCNS-V, MCNE-V, Base, and Sky. In detail, MCNS-V and MCNE-V are both constructed using *Viterbi*, while Base and Sky are built based on the *Base* and *Sky* algorithms, respectively. During the testing process, MCNS-V only provides a given start event while MCNE-V provides start and end events.

Specifically, the above inference datasets all exist in a multiple-choice setting, i.e., the event representation model chooses a positive event from one golden choice and four corrupted choices for the inference at each step.

In addition to the inference ability, we also evaluate the representation ability of our proposed models, i.e., how well similar or dissimilar event pairs can be correctly classified. This representation ability is assessed on the basis of two kinds of similarity tasks: the hard similarity task and the transitive sentence similarity task.

1) *Hard Similarity:* Two types of difficult-to-classify event pairs are considered, namely, semantically close but lexically distant event pairs and semantically distant but lexically overlapping event pairs. Such similar and dissimilar event pairs are present with the same number of samples in the original dataset [9] and have been extended by Ding *et al.* [10] to 1000 event pairs in total. As the evaluation metric, accuracy, i.e., the percentage of correctly classified event pairs, is adopted for the hard similarity task.
2) *Transitive Sentence Similarity:* Similar event pairs are constructed from transitive sentences. Extracted from a transitive similarity dataset [40], each pair of events has been annotated by specialized annotators with a similarity score from 1 to 7. For the evaluation metric, the Spearman's correlation ($\rho \in [-1, 1]$) is adopted to measure how similar the model predictions are to the human annotations.

B. Model Summary

In accordance with the model taxonomy presented in Section II-A, we select the state-of-the-art models as baselines, which are listed as follows. **Event-comp:** an intra-event-based method that considers intra-event elements based on a fully connected network [7]. **Role-factor:** an intra-event-based method that models multiplicative interactions among intra-event elements based on a tensor network [9]. **EventTransE:** an inter-event-based method that explores inter-event relations based on the discourse relations [14]. **SAM-Net:** an event-segment-based method that explores the event-segment relations [16]. **FEEL:** an external-knowledge-based method that introduces the sentiment and animacy information [8]. **IntSent:** an external-knowledge-based method that introduces the intent and sentiment information [10]. **UniFA-S:** a unified fine-tuning-based method that combines three levels of training, namely, intra-event, inter-event and scenario-level training.

Next, we list the models proposed in this article that are considered for comparisons. **HeterEvent$_{[L]}$:** a heterogeneous-graph-based model with a specific loss [L], e.g., the word-level loss (HeterEvent$_{[W]}$), the event-level loss (HeterEvent$_{[E]}$) or the combined loss (HeterEvent$_{[W+E]}$). **MulCL$_{[L]}$:** a pretrained HeterEvent$_{[L]}$ approach following the proposed MulCL framework, again with either the word-level loss (MulCL$_{[W]}$), the event-level loss (MulCL$_{[E]}$) or the combined loss (MulCL$_{[W+E]}$).

C. Model Configuration

Following Granroth-Wilding and Clark [7] and Lee and Goldwasser [8], [14], we choose the New York Times portion of the Gigaword corpus[1] as the raw text corpus. In addition, we use the Stanford CoreNLP tools [41] to extract the dependency information and coreference chains. Based on the coreference chains, we create event chains in the form of $(pred, subj, obj)$. For the extraction of intra-event words, we retain the complete mention spans rather than only headwords. For the detailed extraction process, the reader is referred to Lee and Goldwasser [14]. Finally, we select 1.4 M event chains as the training set, 10 k event chains as the development set and 10 k event chains as the test set. The number of chains in the MCNC, MCNS-V, MCNE-V, Base, and Sky subsets of the test set is consistently set to 2 k. To evaluate the representation ability, the initial representation datasets [9], [10] are directly employed as the test set. We employ a nonparametric classifier (i.e., the cosine similarity) to classify these event pairs.

In the pretraining stage, we perform a grid search from 0 to 1 to obtain the optimal trade-off weights in (6) and (14). Based on the evaluation loss on the validation dataset, we find that the models work well when

[1] https://catalog.ldc.upenn.edu/LDC2003T05

TABLE I

INFERENCE PERFORMANCE IN TERMS OF ACCURACY (%). "ONE-STEP" AND "MULTISTEP" DENOTE THE ONE-STEP AND MULTISTEP INFERENCE TASKS, RESPECTIVELY. THE RESULTS PRODUCED BY THE BEST BASELINE AND THE BEST OVERALL PERFORMER IN EACH COLUMN ARE UNDERLINED AND BOLDFACED, RESPECTIVELY. THE † SYMBOL INDICATES RESULTS PRODUCED BY ZHENG *et al.* [17], AND THE ‡ SYMBOL INDICATES RESULTS FROM ZHENG *et al.* [1]

Model	One-step	Multistep			
	MCNC	MCNS-V	Base	Sky	MCNE-V
Event-comp †	46.3	29.9	27.8	38.4	32.5
Role-factor †	48.8	28.6	28.3	39.6	32.5
EventTransE †	63.7	59.5	51.2	64.5	60.9
SAM-Net †	54.3	46.2	43.2	50.4	49.2
FEEL †	51.6	41.6	38.5	46.0	44.8
IntSent †	56.4	44.7	42.2	49.6	48.5
UniFA-S †	66.3	64.0	**55.4**	67.2	64.5
HeterEvent$_{[W]}$ ‡	62.6	58.7	48.9	63.2	59.1
HeterEvent$_{[E]}$ ‡	63.5	59.8	50.7	65.4	60.8
HeterEvent$_{[W+E]}$ ‡	64.4	60.3	51.3	65.7	61.7
MulCL$_{[W]}$	64.9	61.7	52.8	65.9	62.1
MulCL$_{[E]}$	65.8	63.2	54.5	67.2	63.3
MulCL$_{[W+E]}$	**66.8**	**64.4**	55.0	**67.5**	**65.3**

TABLE II

REPRESENTATION PERFORMANCE IN TERMS OF ACCURACY (%) AND THE SPEARMAN'S CORRELATION (ρ). "ORIG." AND "EXT." REPRESENT THE ORIGINAL AND EXTENDED HARD SIMILARITY TASKS, RESPECTIVELY, WHILE "TRANS. SENTENCE SIMILARITY" DENOTES THE TRANSITIVE SENTENCE SIMILARITY TASK. THE RESULTS PRODUCED BY THE BEST BASELINE AND THE BEST OVERALL PERFORMER IN EACH COLUMN ARE UNDERLINED AND BOLDFACED, RESPECTIVELY. THE † SYMBOL INDICATES RESULTS PRODUCED BY ZHENG *et al.* [17], WHILE THE ♠ SYMBOL MEANS RESULTS REIMPLEMENTED BY OURSELVES

Model	Hard Similarity (Acc. %)		Trans. Sentence
	Orig. [9]	Ext. [10]	Similarity (ρ)
Event-comp †	33.9	18.7	0.57
Role-factor †	43.5	20.7	0.64
EventTransE †	53.7	48.1	0.65
SAM-Net †	51.3	45.2	0.59
FEEL †	58.7	50.7	0.67
IntSent †	77.4	62.8	0.74
UniFA-S †	78.3	64.1	0.75
HeterEvent$_{[W]}$ ♠	76.1	61.3	0.70
HeterEvent$_{[E]}$ ♠	76.8	62.0	0.72
HeterEvent$_{[W+E]}$ ♠	76.6	62.3	0.73
MulCL$_{[W]}$	77.7	63.1	0.73
MulCL$_{[E]}$	**78.5**	63.9	0.75
MulCL$_{[W+E]}$	78.3	**64.3**	**0.76**

λ_1–λ_3 in (6) are set to 0.4, 0.4, and 0.2, respectively, and α_1 and α_2 in (14) are set to 0.7 and 0.3, respectively. For other hyperparameters in the MulCL framework, in our pilot experiments, we find them work well when N_c, N_s and N_g are set to 24, 8, and 8, respectively. During training, we set the batch size to 128 and the regularization weight [i.e., λ in (15) and (16)] to 10^{-5}. We adopt the Adam optimizer [42] with an exponentially descending learning rate to optimize the loss. We also use the gradient clipping with a threshold of 10 to stabilize GRU training [43]. For the word embeddings, we adopt the pretrained BERT-Base-Uncased version to initialize the model; the reader is referred to Devlin *et al.* [28] for details. Other weights or trade-off matrices are initialized using Xavier initialization [44]. Notably, we employ the same parameters in all models for each inference and representation dataset.

D. Overall Performance

1) Inference Ability: We examine the inference ability of our proposed models as well as the baselines for the one-step inference task (i.e., MCNC) and the multistep inference tasks (i.e., MCNS-V, Base, Sky, and MCNE-V). For comparison, we present the experimental results in Table I.

First, we focus on the comparison among the baselines. Among all baselines, the best-evaluated method in terms of its inference performance is UniFA-S [17]. Compared with other methods, UniFA-S shows advantages by virtue of its unified fine-tuning architecture, which constructs a series of self-supervised pretraining tasks to integrate different levels of event training.

Next, we compare our proposals against the baselines. Clearly, the proposals without pretraining (i.e., HeterEvent$_{[W]}$, HeterEvent$_{[E]}$ and HeterEvent$_{[W+E]}$) consistently underperform against the self-supervised UniFA-S method, which indicates that training only on limited labeled data cannot match the benefits offered by self-supervised learning. Despite their failure against UniFA-S, however, our proposals without pretraining still exhibit obvious superiority compared with the baselines without pretraining. Their advantages against these baselines indicate that by considering both homogeneous and heterogeneous relations, the connections between physically disconnected but semantically related nodes can be effectively constructed. Furthermore, when the HeterEvent graph is integrated with the MulCL pretraining framework, a considerable improvement of the inference performance can be achieved. For instance, MulCL$_{[W+E]}$ outperforms the best baseline UniFA-S on four out of five inference tasks (i.e., MCNC, MCNS-V, Sky, and MCNE-V) and keeps the same step with UniFA-S on Base. Compared with the corresponding supervised-only model, MulCL$_{[W+E]}$ can achieve a 5.3% average improvement in terms of accuracy over HeterEvent$_{[W+E]}$ for the inference tasks. These improvements prove the effectiveness of MulCL in pretraining the HeterEvent graph.

2) Representation Ability: Whereas complete event chains are considered in the inference-based tasks, the representation-ability-based tasks rely on only pairs of individual events, which cannot be readily captured by training graph-based models. In other words, pairs of individual events cannot be used to construct an event graph with sufficient interactions.

Intuitively, if we wish to evaluate the representation ability of our proposed graph-based models, we must enrich the testable similar or dissimilar event pairs for the related tasks with narrative event chains. As for other baselines without the graph structure, we also test their representation ability in the same evaluating mode to keep the comparison fair.

Given a testable event pair $\{e_m, e_n\}$, we first retrieve highly word-matched event-chain sets $\mathbb{T}_m$ and $\mathbb{T}_n$ for e_m and e_n, respectively, from the whole training set $\mathbb{T}$ as the "virtual context." The more matching words there are between a retrieved event chain and the testable event pair, the more contextual information for the event pair this event chain contains. Hence, we simply choose the average word-matching score for all events in the event chain as the matching score between the testable event pair and the event chain, which can be formulated as follows:

$$\begin{cases} \mathcal{S}_{e_m,\mathcal{T}_i} = \dfrac{1}{|\mathcal{T}_i|} \displaystyle\sum_{e^* \in \mathcal{T}_i} \mathcal{M}(e^*, e_m), \text{s.t. } \mathcal{T}_i \in \mathbb{T}_m \\ \{\mathcal{S}_{e_m,\mathcal{T}_i}\}_{i=1}^{|(T)_m|} = \text{Normal}\left(\{\mathcal{S}_{e_m,\mathcal{T}_i}\}_{i=1}^{|(T)_m|}\right) \end{cases} \tag{17}$$

where Normal$(\cdot)$ denotes the softmax normalization function, $\mathcal{S}_{e_m,\mathcal{T}_i}$ denotes the probability that the retrieved event chain $\mathcal{T}_i$ can serve as the "virtual context" for the testable event e_m, and $\mathcal{M}(e^*, e_m)$ represents the word-matching score between events e^* and e_m. Given the retrieved event chains and matching scores, we can append the testable event to the end of each retrieved event chain and feed them into our trained graph-based models to obtain the corresponding event representations, i.e., $e(e_m^{\mathcal{T}_i})$, $\mathcal{T}_i \in \mathbb{T}_m$. Then, the final context-rich event representation can be obtained through weighted averaging, i.e.,

$$e^r(e_m) = \frac{1}{|\mathbb{T}_m|} \sum_{\mathcal{T}_i \in \mathbb{T}_m} \mathcal{S}_{e_m,\mathcal{T}_i} \cdot e\left(e_m^{\mathcal{T}_i}\right). \tag{18}$$

For discussed models, we consistently employ the cosine similarity to compute similarity scores for testable event pairs. As the evaluation metric, following Ding *et al.* [10], we adopt the accuracy, which is defined as the percentage of cases in which a similar pair receives a higher cosine value than a dissimilar pair. To facilitate comparison, we present the representation performance of all discussed models in Table II.

Similar to the inference ability evaluation, the self-supervised UniFA-S method still shows a dominant advantage over the other baselines, indicating that appropriately designed self-supervised pretraining tasks can help improve the representation ability. Next, we focus on the comparison between our proposals and the baselines. For the proposals without pretraining, although they cannot outperform the self-supervised UniFA-S, they can achieve a representation performance comparable to that of the external-knowledge-based model IntSent, thus verifying the effectiveness of graph-based context enrichment. The proposals combined with MulCL pretraining can return further performance improvements relative to the pretraining-free proposals. For instance, MulCL$_{[W+E]}$ achieves improvements of 2.2% and 3.1% in terms of accuracy for the original and extended hard similarity tasks, respectively, and an improvement of 4.1% in terms of the Spearman's correlation for the transitive sentence similarity task. Apparently, MulCL$_{[W+E]}$ is the best performer among all discussed models in terms of its representation ability, achieving the best performance for two of the three representation tasks and performing on par with MulCL$_{[E]}$ for the original hard similarity task.

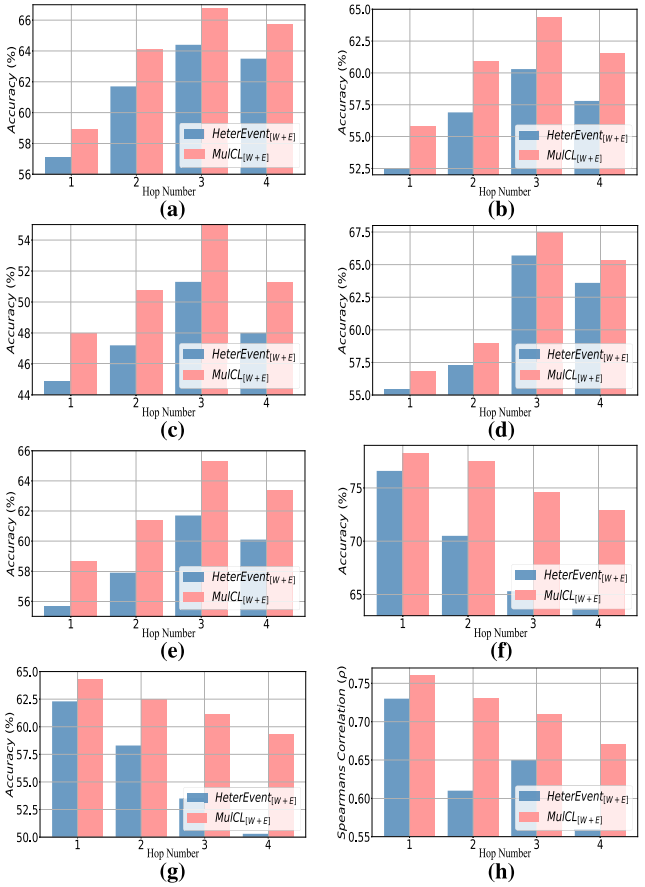


Fig. 3. Relationships between the number of hops (i.e., the number of information-passing layers) and the model performance for HeterEvent$_{[W+E]}$ and MulCL$_{[W+E]}$. (a) MCNC. (b) MCNS-V. (c) Base. (d) Sky. (e) MCNE-V. (f) Original Hard Similarity. (g) Extended Hard Similarity. (h) Transitive Sentence Similarity.

E. Graph Properties

1) Message-Passing Layer: One information-passing layer is equivalent to information aggregation for all one-hop neighborhood nodes. Intuitively, the more information-passing layers there are, the deeper the information interactions among nodes in the heterogeneous graph. However, adding too many information-passing layers will cause graph-based methods to suffer from oversmoothing [37]. Hence, it is meaningful to explore the influence of the number of information-passing layers on the inference performance. In Fig. 3, we present the results of HeterEvent$_{[W+E]}$ and MulCL$_{[W+E]}$ with various numbers of information-passing layers (denoted by l, $l = 1, 2, 3, 4$) for different inference- and representation-based tasks.

First, let us focus on inference-based tasks, i.e., from Fig. 3(a)–(e). We can clearly observe that as the number of layers increases, the performance of either HeterEvent$_{[W+E]}$

or MulCL$_{[W+E]}$ on any task initially increases to a maximum and then drops. In particular, these two models always achieve the best performance when the number of hops is 3. This consistent pattern may be attributed to the size of the graph. Since most HeterEvent graphs have fewer than 50 nodes, the nodes themselves and their three-hop neighborhood nodes can realize minimum node coverage of the graph. Once the number of hops is greater than 3, the oversmoothing problem will be amplified, causing degeneration in the inference performance. In addition, MulCL$_{[W+E]}$ always shows a performance advantage over HeterEvent$_{[W+E]}$ for each number of hops, proving the effectiveness of the MulCL framework.

Next, we focus on the representation-based tasks, i.e., from Fig. 3(f)–(h). Similarly, MulCL$_{[W+E]}$ maintains a performance advantage over HeterEvent$_{[W+E]}$ at different numbers of hops. As the number of hops increases, however, the representation performance of HeterEvent$_{[W+E]}$ and MulCL$_{[W+E]}$ exhibits a different trend than that observed in the case of inference-based tasks. In particular, the performance of both HeterEvent$_{[W+E]}$ and MulCL$_{[W+E]}$ is at its best with one-hop message passing and then immediately starts to decline as the number of hops increases. This difference may be attributable to the "virtual context" event chains, which provide only limited context information for the testable event pairs. As the number of hops increases, more irrelevant and noisy events will be included in the interactions of the testable event pairs.

2) Average Node Degree: For a graph, the average degree of all nodes measures the overall connection level of the whole graph. On the other hand, it can also reflect the closeness between the given event chain and a missing event. Therefore, it is meaningful to explore the impact of the average degree of all nodes on the inference performance.

We first calculate the average node degree for each example based on the corresponding constructed graph. For simplicity, we do not distinguish the degrees of different types of nodes. Based on the distribution of the average node degree, we can roughly divide the test examples into six intervals, i.e., $(0, 4)$, $[4, 4.5)$, $[4.5, 5)$, $[5, 5.5)$, $[5.5, 6)$, and $[6, \infty)$. We group the performances of HeterEvent$_{[W+E]}$ and MulCL$_{[W+E]}$ on the inference and representation-based tasks in accordance with these intervals and present the results in Fig. 4. From Fig. 4, we can clearly observe that for any task, HeterEvent$_{[W+E]}$ and MulCL$_{[W+E]}$ both obtain a stable performance boost in terms of accuracy as the average node degree increases. This uniform behavior proves that a higher average node degree reflects stronger inference and representation capabilities of the HeterEvent graph. In other words, the higher the overall connection level of the graph is, the easier it is to realize node interaction to facilitate inference. Additionally, MulCL$_{[W+E]}$ always shows a performance advantage against HeterEvent$_{[W+E]}$ in each interval of the average node degree. As the average node degree increases, however, the advantage gap between MulCL$_{[W+E]}$ and HeterEvent$_{[W+E]}$ gradually shrinks. The collapse of the advantage gap may be explained by the fact that the pretraining process can provide more supervised signals for a supervision-scarce HeterEvent graph with a low average node degree, while for a supervision-abundant HeterEvent graph with a high average node degree, the benefit of the extra supervised signals begins to vanish.

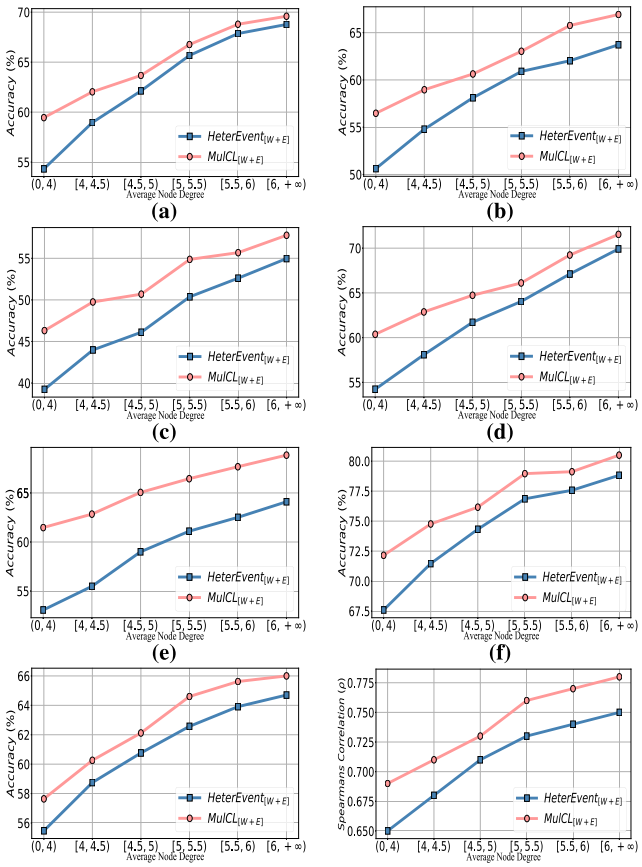


Fig. 4. Relationships between the number of hops (i.e., the number of information-passing layers) and the model performance for HeterEvent$_{[W+E]}$ and MulCL$_{[W+E]}$. (a) MCNC. (b) MCNS-V. (c) Base. (d) Sky. (e) MCNE-V. (f) Original Hard Similarity. (g) Extended Hard Similarity. (h) Transitive Sentence Similarity.

TABLE III

ABLATION STUDIES OF MULCL$_{[W+E]}$ ON INFERENCE TASKS. THE LARGEST DROP IN EACH COLUMN IS MARKED WITH THE ↓ SYMBOL

Model	One-step	Multistep Inference			
	MCNC	MCNS-V	Base	Sky	MCNE-V
MulCL$_{[W+E]}$	66.8	64.4	55.0	67.5	65.3
−Order Contrast	66.5	64.0	54.9	67.3	65.1
−Replacement Contrast	66.6	64.1	54.7	67.3	65.0
−Composition Contrast	65.9	63.5	54.1	66.9	64.7
−Normal–Corrupted Contrast	65.0	61.3	52.0	66.1	62.5
−Corrupted–Corrupted Contrast	64.5 ↓	60.7 ↓	51.7 ↓	65.9 ↓	62.2 ↓

F. Ablation Studies

To better understand the contributions of the different modules to the model performance, we conduct ablation studies using our proposed MulCL$_{[W+E]}$ model on these discussed inference-and-representation ability-based tasks. In the ablation studies, we remove or replace certain specific layers or

TABLE IV

ABLATION STUDIES OF MULCL$_{[W+E]}$ ON REPRESENTATION TASKS. THE LARGEST DROP IN EACH COLUMN IS MARKED WITH THE ↓ SYMBOL

Model	Hard Similarity (Acc. %)		Trans. Sentence
	Orig. [9]	**Ext.** [10]	**Similarity** (ρ)
MulCL$_{[W+E]}$	78.3	64.3	0.76
−Order Contrast	77.0	62.8	0.74
−Replacement Contrast	77.3	63.0	0.75
−Composition Contrast	76.8↓	62.5↓	0.73↓
−Normal–Corrupted Contrast	77.9	63.8	0.76
−Corrupted–Corrupted Contrast	77.4	63.5	0.75

modules to explore their influence on our proposed model, which is represented by the notation "-." For example, "–Order contrast" denotes the direct removal of the order contrast loss in (6). Following this setting, the associated performance results on the inference and representation-based tasks are shown in Tables III and IV, respectively.

Clearly, the elimination of any contrast loss causes performance degeneration on all inference- and representation-based tasks, thereby proving the effectiveness of these contrast components. Now, let us focus on the comparison between the SeqCL and GraCL components in terms of performance degeneration. For the inference-based tasks, removing sequential-view contrast losses results in relatively slight performance drops, while the removal of graph-view contrast losses causes more obvious performance deterioration. The largest drop occurs with the removal of the corrupted–corrupted contrast loss, which indicates that pretraining the graph structure can help improve the inference ability of the event representation model. For the representation-based tasks, the differences in the performance drops show the opposite trend: the elimination of sequential-view contrast losses leads to greater performance decay than the removal of graph-view contrast losses. This discrepancy can be explained by the fact that representation or inference ability benefits from different views of structures: pretraining the graph structure is helpful to the inference ability while pretraining the sequential structure boosts the representation ability.

V. CONCLUSION AND FUTURE WORK

In this article, we introduce a novel heterogeneous event graph network (HeterEvent) for event representation. However, data sparsity and the scarcity of supervised data make HeterEvent training difficult. To alleviate the associated risks, we design a MulCL for pretraining HeterEvent from two structural perspectives (i.e., the sequential perspective and the graph perspective) without the participation of any external knowledge or labeled data. Extensive experiments validate the effectiveness of MulCL for pretraining the HeterEvent graph. In particular, considering sequential-view contrasts can help improve the representation ability of HeterEvent, while considering graph-view contrasts can boost the inference ability.

Regarding future work, we continue to research on the application of unlabeled data. On the one hand, we plan to employ semi-supervised learning methods to generate pseudo labels for unlabeled event chains, which can further support the subsequently supervised training. Such self-training paradigm can promote the utility of unlabeled data. On the other hand, we investigate to employ the *prompt* [45] to help generate missing events in incomplete event chains, which can transfer the knowledge from pretraining language model to assist the downstream tasks without fine-tuning too much parameters.

ACKNOWLEDGMENT

All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors. An earlier version of this article appeared in the Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020) [1]. In this extension, we attempt to address the particular issues of sparsity and undertraining in event representation training by proposing a multistructure contrastive learning framework. In addition, we include more baselines and research tasks to evaluate our proposals.

REFERENCES

[1] J. Zheng, F. Cai, Y. Ling, and H. Chen, "Heterogeneous graph neural networks to predict what happen next," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 328–338.

[2] R. C. Schank and R. P. Abelson, "Scripts, plans, goals, and understanding: An inquiry into human knowledge structures," Lawrence Erlbaum Associates, Mahwah, NJ, USA, Tech. Rep., 1977.

[3] N. Chambers and D. Jurafsky, "Unsupervised learning of narrative event chains," in *Proc. ACL*, 2008, pp. 789–797.

[4] N. Chambers and D. Jurafsky, "Unsupervised learning of narrative schemas and their participants," in *Proc. ACL*, 2009, pp. 602–610.

[5] X. Liu *et al.*, "Self-supervised learning: Generative or contrastive," *CoRR*, vol. abs/2006.08218, Jun. 2021.

[6] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph contrastive learning with adaptive augmentation," *CoRR*, vol. abs/2010.14945, Apr. 2020.

[7] M. Granroth-Wilding and S. Clark, "What happens next? Event prediction using a compositional neural network model," in *Proc. AAAI*, 2016, pp. 2727–2733.

[8] I. Lee and D. Goldwasser, "FEEL: Featured event embedding learning," in *Proc. AAAI*, 2018, pp. 4840–4847.

[9] N. Weber, N. Balasubramanian, and N. Chambers, "Event representations with tensor-based compositions," in *Proc. AAAI*, 2018, pp. 4946–4953.

[10] X. Ding, K. Liao, T. Liu, Z. Li, and J. Duan, "Event representation learning enhanced with external commonsense knowledge," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 4893–4902.

[11] K. Pichotta and R. Mooney, "Statistical script learning with multi-argument events," in *Proc. 14th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2014, pp. 220–229.

[12] Z. Wang, Y. Zhang, and C.-Y. Chang, "Integrating order information and event relation for script event prediction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 57–67.

[13] Z. Li, X. Ding, and T. Liu, "Constructing narrative event evolutionary graph for script event prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 4201–4207.

[14] I.-T. Lee and D. Goldwasser, "Multi-relational script learning for discourse relations," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 4214–4226.

[15] R. Prasad *et al.*, "The Penn discourse TreeBank 2.0," in *Proc. LREC*, 2008.

[16] S. Lv, W. Qian, L. Huang, J. Han, and S. Hu, "SAM-Net: Integrating event-level and chain-level attentions to predict what happens next," in *Proc. AAAI*, 2019, pp. 6802–6809.

[17] J. Zheng, F. Cai, and H. Chen, "Incorporating scenario knowledge into a unified fine-tuning architecture for event representation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 249–258.
[18] M. Sap *et al.*, "ATOMIC: An atlas of machine commonsense for if-then reasoning," in *Proc. AAAI*, 2019, pp. 3027–3035.
[19] R. D. Hjelm *et al.*, "Learning deep representations by mutual information estimation and maximization," in *Proc. ICLR*, 2019.
[20] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9726–9735.
[21] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn. (PMLR)*, vol. 119, 2020, pp. 1597–1607.
[22] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. CVPR*, 2006, pp. 1735–1742.
[23] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, Jul. 2018.
[24] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. ECCV*, vol. 12356, 2020, pp. 776–794.
[25] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *Proc. ICLR*, 2019.
[26] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, vol. 9, 2010, pp. 297–304.
[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[28] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.
[29] L. Kong, C. de Masson d'Autume, L. Yu, W. Ling, Z. Dai, and D. Yogatama, "A mutual information maximization perspective of language representation learning," in *Proc. ICLR*, 2020.
[30] D. Iter, K. Guu, L. Lansing, and D. Jurafsky, "Pretraining with contrastive sentence objectives improves discourse performance of language models," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4859–4870.
[31] Y. Qu, D. Shen, Y. Shen, S. Sajeev, J. Han, and W. Chen, "CoDA: Contrast-enhanced and diversity-promoting data augmentation for natural language understanding," *CoRR*, vol. abs/2010.08670, Oct. 2020.
[32] Y. Zhu *et al.*, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 19–27.
[33] D. Graff, J. Kong, K. Chen, and K. Maeda, "English gigaword," *Linguistic Data Consortium*, vol. 4, no. 1, p. 34, 2003.
[34] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[35] R. Socher *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. ACL*, 2013, pp. 1631–1642.
[36] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. ICLR*, 2018.
[37] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2017.
[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[39] M. Tu, G. Wang, J. Huang, Y. Tang, X. He, and B. Zhou, "Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2704–2713.
[40] D. Kartsaklis and M. Sadrzadeh, "A study of entanglement in a categorical framework of natural language," in *Proc. QPL*, vol. 172, 2014, pp. 249–261.
[41] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proc. ACL*, 2014, pp. 55–60.
[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.
[43] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. ICML*, vol. 28, 2013, pp. 1310–1318.
[44] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J. Mach. Learn. Res.*, vol. 9, pp. 249–256, May 2010.
[45] T. Schick and H. Schütze, "Exploiting cloze-questions for few-shot text classification and natural language inference," in *Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2021, pp. 255–269.

Jianming Zheng received the B.S. and M.S. degrees from the National University of Defense Technology, Changsha, China, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the College of Systems Engineering.

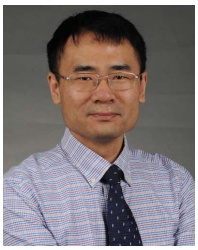
He has published several papers in conference proceedings and journals, such as SIGIR, WWW, CIKM, COLING, and IPM. His research interests include semantics representation and few-shot learning.

Dr. Zheng also serves as a PC Member for AAAI.

Fei Cai received the Ph.D. degree in computer science from the University of Amsterdam, Amsterdam, The Netherlands, in 2016, under the supervision of Prof. Maarten de Rijke.

He is currently an Associate Professor with the National University of Defense Technology, Changsha, China. He has published several papers in WWW, SIGIR, CIKM, FnTIR, TOIS, and TKDE. His research interests include information retrieval and query formulation.

Dr. Cai also serves as a PC Member for KDD, WWW, SIGIR, WSDM, CIKM, and RecSys; and a Reviewer for top journals, such as TKDE, TOIS, TKDD, VLDBJ, IPM, and JASIST.



Jun Liu (Senior Member, IEEE) received the B.S. degree in computer science and technology and the Ph.D. degree in systems engineering from Xi'an Jiaotong University, Xi'an, China, in 1995 and 2004, respectively.

He is currently a Professor with the Department of Computer Science, Xi'an Jiaotong University. He has authored more than 90 research papers in various journals and conference proceedings. His research interests include NLP, CV, and e-learning.

Dr. Liu has won the Best Paper Award at IEEE ISSRE 2016 and IEEE ICBK 2016. He has been serving as an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS) since 2020 and has served as a Guest Editor for many technical journals, such as *Information Fusion* and IEEE SYSTEMS JOURNAL. He also acted as the conference/workshop/track chair at numerous conferences.

Yanxiang Ling received the M.S. degree in information systems engineering from the National University of Defense Technology, Changsha, China, in 2013, where she is currently pursuing the Ph.D. degree.

She has published several articles in WWW, SIGIR, and IPM. Her research interests include neural response generation, question generation, and information retrieval.

Honghui Chen received the Ph.D. degree in operational research from the National University of Defense Technology, Changsha, China, in 2007.

He is currently a Professor with the National University of Defense Technology. He has published several articles in SIGIR, IPM, and other top journals. His research interests include information systems and information retrieval.