Event Representation Learning with Multi-Grained Contrastive Learning and Triple-Mixture of Experts

Tianqi Hu, Lishuang Li*, Xueyang Qin, Yubo Feng
School of Computer Science and Technology, Dalian University of Technology
Dalian,China
htqnlp@gmail.com, lilishuang314@163.com, qinxueyang@snnu.edu.cn, argmax@126.com

Abstract

Event representation learning plays a crucial role in numerous natural language processing (NLP) tasks, as it facilitates the extraction of semantic features associated with events. Current methods of learning event representation based on contrastive learning processes positive examples with single-grain random masked language model (MLM), but fall short in learning information inside events from multiple aspects. In this paper, we introduce multi-grained contrastive learning and triple-mixture of experts (**MCTM**) for event representation learning. Our proposed method extends the random MLM by incorporating a specialized MLM designed to capture different grammatical structures within events, which allows the model to learn token-level knowledge from multiple perspectives. Furthermore, we have observed that mask tokens with different granularities affect the model differently, therefore, we incorporate mixture of experts (MoE) to learn the importance weights associated with different granularities. Our experiments demonstrate that MCTM outperforms other baselines in tasks such as hard similarity and transitive sentence similarity, highlighting the superiority of our method.

Keywords: contrastive learning, mixture of experts, event representation learning

1. Introduction

Events are one of the most common objective entities in People's daily life. Structural events and learning their representations have played an important role in the development of the field of NLP (Li et al., 2018b). The goal of event representation learning is to learn from the text how to convert events into a form that computers can understand and process (such as event embeddings), so as to better support NLP and related applications. By acquiring distributed representations of events, we can construct a semantic model of events within computers, enabling them to comprehend the significance, interpretation, and role of events across various scenarios. This comprehension enhances various downstream tasks, including successive events generation (Martin et al., 2018), event detection (Deng et al., 2021), event prediction (Granroth-Wilding and Clark, 2016) and story generation (Chen et al., 2021).

Previous studies (Lee and Goldwasser, 2019) applied statistical script learning methods to embed event representations and classify the relationship between events based on similarity. However, relying solely on the co-occurrence relationship of events to infer similarity fails to capture the detailed hidden feature information between events. Some recent studies have focused on incorporating external knowledge bases, such as external commonsense knowledge (Ding et al., 2019), human action intentions (Ding et al., 2019) and sentiments (Sap et al., 2019) to provide finer granularity. Furthermore, there have been attempts (Vijayaraghavan and Roy, 2021) to continuously update the feature embeddings of events by combining social knowledge, commonsense knowledge, and continuous learning. However, these methods involve a wide range of knowledge, making it challenging to encompass them all. Recent work (Gao et al., 2022) considers using contrastive learning to learn event representations in co-occurring events, which has shown promising results. However, the single and random masking method used in this approach limits the model's ability to learn the internal information of events from multiple perspectives. We observe that although the masked language model (MLM) objective loss serves as an auxiliary loss, its impact on the final model performance cannot be ignored.

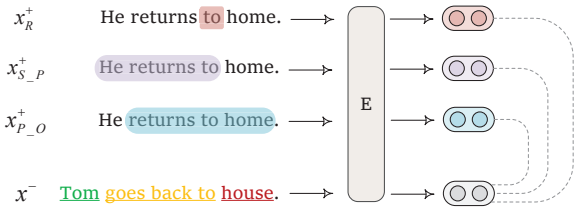


Figure 1: Illustration of the multi-grained positive mask labeling method for contrastive learning. E is the encoder of the model. $x_R^+, x_{S_P}^+, x_{P_O}^+$ are positive examples, and x^- is the corresponding negative example.

As depicted in Figure 1, when using the previous common single-grained labeling method (Gao et al., 2022), where any token in the event is ran-

*Corresponding author

LREC-COLING 2024, pages 6643–6654

domly masked, we obtain limited knowledge compared to the negative example (x^-). For instance, when "*to*" is randomly masked in x_R^+, there is no meaningful similar or opposite semantic structure compared to the negative example. Therefore, we propose the addition of two additional granularities for mask labeling: subject-predicate and predicate-object. This decision is primarily motivated by the fact that most of the semantic information in simple events resides in these two grammatical structures. As shown in Figure 1, we can observe semantic similarity between "*He returns to*" in x_{SP}^+ and "*He goes back to*" in x^-, as well as between "*returns to home*" in x_{PO}^+ and "*goes back to house*" in x^-. Moreover, different granularities hold varying degrees of importance for the model. Hence, it is crucial to assign greater impact to the granularities that possess higher significance and influence during the model's training process. This necessitates assigning different importance weights to different granularities. To achieve this, we incorporate the Mixture of Experts (MoE) (Shazeer et al., 2017) framework, which enables the model to learn the optimal weights associated with each granularity.

To this end, we propose MCTM: a Multi-Grained Contrastive Learning by using a Triple-Mixture of Experts for event representation learning. Our method adopts a multi-granularity labeling approach for positive examples. In addition, we also use three Mixtures of Expert (MoE) layers to parallelize the model structure, distribute the weight of each granularity. To summarize, our contribution is two-fold:

- We propose MCTM, which leverages multi-grained labels in the positive examples of contrastive learning, enabling an understanding of deep event features from multiple perspectives.
- We adopt a triple-Mixture of Experts layer structure to optimize the model structure so that the model can independently learn the importance weights of each label granularity to achieve better results.

2. Preliminary

In this section, we introduce the preliminary in 4 aspects: Event Presentation Model, Contrastive Learning, Mixture of Experts, and Data Augmentation.

2.1. Event Representation Model

Previous research has predominantly relied on tensor neural networks (NTNs) (Socher et al., 2013) for representing events. NTNs employ a word embedding model to convert individual words into vectors and generate a three-dimensional tensor. However, this approach has limitations, such as the need for extensive annotation and the inability to handle events with multiple additional elements, such as location and timing words. Since the release of BERT (Devlin et al., 2019), researchers have increasingly considered pre-trained language models as a replacement for static word representations. BERT offers advantages such as flexible event representation and portability. Consequently, we also utilize BERT as our backbone model. In the following sections, we discuss the composition of events and the precoding method.

Simple events are broadly defined in the form:

$$Event = (Subject, Predicate, Object). \quad (1)$$

The BERT encoder can process text and output a sequence of tokens in a fixed format (a piece of text starts with [CLS] and ends with [SEP] after encoding). So the event tokens will be expressed as follows after being input into BERT:

$$Event_{tokens} = [CLS], Sub, Pre, Obj, [SEP]. \quad (2)$$

The input sequence is obtained after converting each word (including [CLS] and [SEP]) to the corresponding ID. Assuming the input sequence is $\boldsymbol{x} = [x_{CLS}, x_1, ..., x_n, x_{SEP}]$, then BERT will eventually return the tensor of this set of sequences, as follows:

$$BERT_{\boldsymbol{x}} = [\boldsymbol{v}_{CLS}, \boldsymbol{v}_{x_1}, \cdots, \boldsymbol{v}_{x_n}, \boldsymbol{v}_{SEP}], \quad (3)$$

where $\boldsymbol{v}_{CLS}$ and $\boldsymbol{v}_{SEP}$ are the representations for [CLS] and [SEP] tokens. Moreover, $\boldsymbol{v}_{x_n}$ represents the representation of each word in the text.

2.2. Contrastive Learning

Event representation learning aims to abstract events in natural language text into mathematical representations. The challenge of this problem is that the same event can appear in different forms in different texts, so a method is needed to capture the commonality and variability of events.

Contrastive learning provides an effective way to address this problem. In event representation, the basic idea of contrastive learning is to compare the similarities or differences between different text segments. By comparing the representations of the same event in different texts, contrastive learning can capture the commonality of events and generalize across different texts. Additionally, contrastive learning can also use negative samples to capture the variability of events. Therefore, using contrastive learning can bring similar events closer and push irrelevant events farther away.

Recently, InfoNCE objective (Oord et al., 2018), as a powerful contrastive learning loss function,

has an excellent effect in the field of representation learning. The core idea of this loss function is to use the mutual information in information theory to measure the correlation between two representations learned by the model, thereby facilitating the learning of more discriminative representations.

The original InfoNCE considers single positive and negative examples, here we improve it to multiple positive and negative examples to achieve better results. Suppose given M paired event examples $\mathcal{D} = \{(\boldsymbol{x_i}, \boldsymbol{x_i^+})\}_{i=1}^{M}$, where $\boldsymbol{x_i^+}$ is a positive instance for $\boldsymbol{x_i}$, also they are semantically related. We follow the contrastive framework in (Gao et al., 2022), and the training objective for $(\boldsymbol{x_i}, \boldsymbol{x_i^+})$ is presented in a softmax form with mini-batch negatives.

$$\mathcal{L} = -\log \frac{g(c_i, c_i^+)}{g(c_i, c_i^+) + \sum_{k \in \mathcal{M}(i)} g(c_i, c_k)}, \quad (4)$$

where c_i and c_i^+ denote the representation of $\boldsymbol{x_i}$ and $\boldsymbol{x_i^+}$, respectively. $k \in \mathcal{M}(i)$ is the index of mini-batch negatives.

The function $g(\alpha, \beta)$ in Formula 4 calculates the similarity between α and β. The numerator part represents the similarity between positive examples, and the denominator represents the similarity between positive and negative examples. Therefore, the greater the similarity of the same category, the smaller the similarity of different categories, and the smaller the loss. The formula of the function g is

$$g(c_i, c_k) = \exp(\frac{c_i^T c_k}{\tau}), \quad (5)$$

where τ is the temperature coefficient. τ is used to control the model's discrimination against negative samples. We determine the value of this hyperparameter after specific experiments.

2.3. Mixture of Experts

Since MoE (Mixture of Experts) was first proposed in Jacobs et al. (1991); Jordan and Jacobs (1994), it has been the subject of much research (Jacobs et al., 1991; Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2021; Du et al., 2022; Xue et al., 2022). As a combined model, MoE is different from general neural networks in that it separates and trains multiple models (experts) based on data. MoE as a layer rather than a whole model, it consists a set of n "expert networks" ($E_N = \{E_1, E_2 \cdots, E_n\}$) and a "gating network" (G) whose output is a $n - dimension$ vector. Assuming given an input x, we denote $G_i(x)$ and $E_i(x)$ as the output of the gating network and the output of the i-th expert network. We also add gating loss ($\mathcal{L}_G$) and expert loss ($\mathcal{L}_E$) to the final loss function.

2.4. Data Augmentation

For NLP tasks, the data augmentation methods for language representation usually have the following types: synonym replacement, random insertion, random deletion, random swap, and random perturbation. However, some studies in recent years (Gao et al., 2022, 2021) have used *dropout noise* as a data enhancement method and have also proved the effectiveness of this method through experiments. Suppose given $\boldsymbol{x_i}$, we input the same $\boldsymbol{x_i}$ to the encoder with the parametric weights θ twice. Thus we get two embeddings with different dropout masks ($\boldsymbol{c_i}$ and $\boldsymbol{c_i^+}$).

$$c_i = f_\theta(x_i, \omega), \quad (6)$$

$$c_i^+ = f_\theta(x_i, \omega^*) \quad (7)$$

where ω and ω^* are two different random masks for dropout.

3. Our Approach

In this section, we will introduce our approach in detail. Figure 2 presents an overview of our proposed approach.

3.1. Multi-Grained Contrastive Learning

Following Gao et al. (2022), we build our approach on the weakly supervised contrastive framework with the InfoNCE objective. Considering that the previous studies have paid more attention to how the positive examples affect each other, and improved the impact of the number of positive examples on the final model. In order to obtain more information about relationships within and between events, we focus on token-level knowledge. Therefore, we add the objective of multi-granularity masked language modeling (MLM) (Devlin et al., 2019) on the original basis.

For the selection of mask marks, most of the studies generally choose random marks (Zhang et al., 2021; Gao et al., 2022, 2021), that is to say, the knowledge of the token level is not considered for the input text data. We use three granular mask labeling methods to mine deeper event information, namely *random labeling*, *subject-predicate labeling*, and *predicate-object labeling*. We give several concrete examples in Table 1. Random granularity (x_R^+) means that we randomly sample a word as a mask token, which is what most work does. Subject-predicate ($x_{S\_P}^+$) labeling and predicate-object ($x_{P\_O}^+$) labeling means that we mark the subject and predicate, predicate and object in the event as masks respectively.

As mentioned in Section 2.2, $\mathcal{M}(i)$ is the index of mini-batch negatives. How the method differs from InfoNCE is in the construction of the positive

$\boldsymbol{x}$	He returns to home.
$\boldsymbol{x}_{tokens}$	[CLS] He returns to home [SEP]
$\boldsymbol{x}_R^+$	[CLS] He returns [MASK] home [SEP]
$\boldsymbol{x}_{S\_P}^+$	[CLS] [MASK] [MASK] [MASK] home [SEP]
$\boldsymbol{x}_{P\_O}^+$	[CLS] He [MASK] [MASK] [MASK] [SEP]

Table 1: Demonstration of multi-grained labeling on concrete examples, where $\boldsymbol{x}_R^+$, $\boldsymbol{x}_{S\_P}^+$ and $\boldsymbol{x}_{P\_O}^+$ represent random labeling, subject-predicate labeling and predicate-object labeling, respectively.

set $\mathcal{M}(i)$ for $\boldsymbol{x}_i$. We improve InfoNCE objective based on our method. The difference from InfoNCE is that we have added three different granularity calculations to the processing of positive examples, taking token-level knowledge into account. In our method, we generalize Equation 4 to support multi-grained positives learning:

$$\mathcal{L} = \sum_j^3 -\log \frac{g(c_i, c_{i_j}^+)}{g(c_i, c_{i_j}^+) + \sum_{k \in \mathcal{M}(i)} g(c_i, c_k)}, \quad (8)$$

where j denotes three different grain sizes. We calculate the similarity with the anchor points (c_i) from these three perspectives and learn the representation of the event. Moreover, $k \in \mathcal{M}(i)$ is the index of mini-batch negatives (x^-). During the experiment, we found that we could not observe the impact of each granularity on the final result. Therefore, we introduce MoE to solve the problem.

3.2. Combination with MoE

In this subsection, we give a schematic diagram of the specific model (Figure 2) and explain the details of our method.

The experts themselves are neural networks, each with their own parameters. In our initial investigations for this paper, we limit them to feed-forward networks with identical architectures but separate parameters. We define each expert as an MLP:

$$MLP = \{Linear, ReLu, Softmax\}, \quad (9)$$

where the input and output specifications of the Linear layer are the same as those in MoE.

The n experts in the figure are divided into three parts, and each part calculates the weight of different granularity. For example, from 1 to r to calculate the first granularity, each expert will output a weight, and finally the weight of the entire first granularity will be calculated. We will focus on the calculation process of weights and functions in our model.

We define $G(x)$ and $E_i(x)$ as the output of the gating network and the output of the i-th expert network for given input x. Therefore, the outputs of the three parts above-mentioned can be calculated respectively as follows:

$$E(x)_R = \sum_{\alpha=1}^{r} W_\alpha \cdot O(\alpha, x), \quad (10)$$

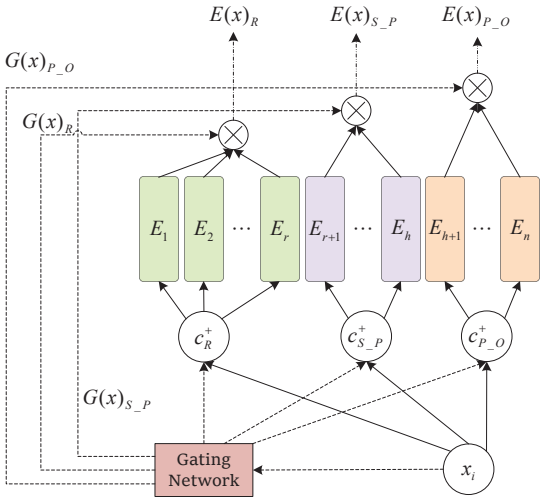


Figure 2: The architecture of the proposed framework. Given an input event $\boldsymbol{x_i}$, we get $\boldsymbol{c}_R^+$, $\boldsymbol{c}_{S\_P}^+$, and $\boldsymbol{c}_{P\_O}^+$ through three different granularity marks, and then enter n experts ($E_1 \dots E_n$), and finally get the weights assigned by the model to different granularities. Moreover, $\boldsymbol{c}_R^+ = f_\theta(R(\boldsymbol{x_i}), \omega^*)$, where $R(x)$ represents random granularity marking of the original event.

$$E(x)_{S\_P} = \sum_{\alpha=r+1}^{h} W_\alpha \cdot O(\alpha, x), \quad (11)$$

$$E(x)_{P\_O} = \sum_{\alpha=h+1}^{n} W_\alpha \cdot O(\alpha, x), \quad (12)$$

where α is the serial number of the experts participating in the operation, the output of each expert α is $O(\alpha, x)$, and W_α is trainable weight matric.

Instead of the simple selection used by non-sparse gating functions (Jordan and Jacobs, 1994), we add sparsity and noise to the softmax gating network. Specifically, we introduce the *Noisy Top-K Gating* mechanism, by sampling and sorting the weights of each expert, and then selecting the top k models with higher weights to participate in the calculation (while the rest will be set to $-\infty$). In this way, the experts who are most useful to the input samples can be reserved as much as possible to participate in the calculation while maintaining the saving of computing resources.

$$G(x) = Softmax(GetTop(Q(x), k)) \quad (13)$$

$$Q_i(x) = (x \cdot W_g)_i + stdnorm() \cdot softplus(x \cdot W_q) \quad (14)$$

Both W_g and W_q in Equation 14 are trainable weight matrices, and we use a standard Gaussian distribution $stdnorm()$ to format the non-linear activation function.

3.3. Training

The final training goal of the model is as follows:

$$\mathcal{L}_{Final} = \varphi(\mathcal{L}_{info} + \mathcal{L}_{mlm}) + \gamma \mathcal{L}_E, \quad (15)$$

where φ and γ are editable hyperparameters. $\mathcal{L}_{info}$ is the modified InfoNCE loss, $\mathcal{L}_{mlm}$ is the masked language model loss. Based on the original InfoNCE, we added the parameter ε of each positive example learning. Finally, $\mathcal{L}_E$ is the expert loss we get from the MoE layer. Below we give some specific calculation procedures.

When we train multi-granularity label comparison learning, the space occupied by the loss function introduced at the beginning is too high, and we could not clarify the impact of each granularity on the model, so we make modifications here.

$$\mathcal{L}_{info} = \sum_{j=1}^{3} -\log \frac{\varepsilon_j g(c_i, c_{ij}^+)}{g(c_i, c_{ij}^+) + \sum_{k \in \mathcal{M}(i)} g(c_i, c_k)} \tag{16}$$

where ε_j represents the importance weights of each granularity, and these three parameters are automatically obtained during the training process of the model. For example, the weight calculation of the three granularities are as follows:

$$\varepsilon_1 = \sum_{i=1}^{r} Q_i(x) \tag{17}$$

$$\varepsilon_2 = \sum_{i=r+1}^{k} Q_i(x) \tag{18}$$

$$\varepsilon_3 = \sum_{i=k+1}^{n} Q_i(x) \tag{19}$$

Moreover, the MLM loss $\mathcal{L}_{mlm}$ is obtained by calculating the cross entropy separately at three different granularities and adding them together. We show Equation 20 below, $\mathcal{L}_{R\_mlm}$, $\mathcal{L}_{S\_P\_mlm}$ and $\mathcal{L}_{P\_O\_mlm}$ represent random granularity, subject-predicate granularity and predicate-object granularity respectively.

$$\mathcal{L}_{mlm} = \mathcal{L}_{R\_mlm} + \mathcal{L}_{S\_P\_mlm} + \mathcal{L}_{P\_O\_mlm} \tag{20}$$

In addition, in the experiment, we have found that the gating network tends to converge to a certain state where it always produces large weights for the same few experts. We analyze that the reason for this problem is that the favored experts are trained faster, which leads to more selection of such experts by the gating network. So we introduce a loss L_I for encouraging all experts to have equal importance and another loss L_L to ensure balanced loads.

$$\mathcal{L}_I(X) = \omega_I \cdot CV(\sum_{x \in X} G(x))^2, \tag{21}$$

$$\mathcal{L}_L(X) = \omega_L \cdot CV(\sum_{x \in X} P(x, i))^2, \tag{22}$$

$$\mathcal{L}_E = \mathcal{L}_I(X) + \mathcal{L}_L(X), \tag{23}$$

where $CV(X)$ is the Coefficient of Variation, which is used to measure the degree of dispersion between samples.

4. Experimental Settings

Following previous studies (Ding et al., 2019; Li et al., 2018a; Lee and Goldwasser, 2019), we evaluate the event representation learning models on two event similarity tasks, a transfer task and a script event prediction downstream task (Lee and Goldwasser, 2019).

The event triples we use for the training data are extracted from the New York Times Gigaword Corpus using the Open Information Extraction system Ollie (Mausam et al., 2012). Our training dataset consists of an extensive collection of 4,029,877 event triplets. As for downstream tasks, we use the MCNC dataset adopted in Lee and Goldwasser (2019)[1] for the downstream task.

We use the Texar-Pytorch package (Hu et al., 2019) to build the model and take BERT (Devlin et al., 2019) as the backbone model. We train our model with a batch size of 256 using an Adam optimizer. The learning rate is set as 2e-7 for the event representation model. The training epochs are set as 3. Some other hyperparameter settings: The number of experts is set as 9 and K is set as 2. In Eq.15: φ is set to 0.48 and γ is set to 0.52. In Eq.21 and Eq.22, ω_I is set to 0.01 and ω_L is set to 0.1, respectively.

4.1. Event Similarity Tasks

Similarity tasks are often a general measure of how good a vector representation is. (Weber et al., 2018) introduce two event related similarity tasks, respectively *Hard Similarity* Task and *Transitive Sentence Similarity*.

Hard Similarity Task The task aims to measure the similarity between two text fragments. Each event pair in the test set contains two groups of events, one group of events has little overlap in vocabulary, but they are similar in semantics, and the other group has a lot of overlap in vocabulary, but the semantics they express are quite different. This dataset contains a total of 115 event groups and 230 pairs of events (Weber et al., 2018) (denoted as "Original Hard Similarity Task").

To evaluate the robustness of event representation, (Ding et al., 2019) extend the above dataset to 1000 event pairs (similar and dissimilar events each account for 50%) (denoted as "Extend Hard Similarity Task"). We use Accuracy as the evaluation metric, which measures the percentage of cases where the similar pair receives a higher cosine value than the dissimilar pair.

Transitive Sentence Similarity We also test the effectiveness of our method on the transitive sentence similarity task (Kartsaklis and Sadrzadeh,

[1] https://github.com/doug919/multi_relational_script_learning

Model	Original hard Sim.(%)	Extend hard Sim.(%)	Transitive sentence Sim.(ρ)
Predicate Tensor (Weber et al., 2018)	41.0	25.6	0.63
Role-factor Tensor (Weber et al., 2018)	43.5	20.7	0.64
SAM-Net (Lv et al., 2019)	51.3	45.2	0.59
KGEB (Ding et al., 2016)	52.6	49.8	0.61
FEEL (Lee and Goldwasser, 2019)	58.7	50.7	0.67
NTN-IntSent (Ding et al., 2019)	77.4	62.8	0.74
UniFA-S (Zheng et al., 2020)	78.3	64.1	0.75
SWCC (Gao et al., 2022)	80.9	72.1	0.82
MCTM (ours)	**81.7**	**75.2**	**0.85**

Table 2: The overall performance on the event similarity task. The best results are bolded. Sim. denotes similarity.

2014), which contains 108 pairs of transitive sentences: short phrases containing a single subject, object and verb (e.g., agent sell property). Every pair is annotated by a human with a similarity score from 1 to 7, and a higher score indicates that the two events are more similar. A larger score indicates that the two events are more similar. Following previous work (Weber et al., 2018; Ding et al., 2019; Gao et al., 2022), we evaluate using Spearman's correlation of the cosine similarity predicted by each method and the annotated similarity score.

4.2. Downstream Task

We also validate the effectiveness of our method on downstream tasks. The Multiple Choice Narrative Cloze (MCNC) task (Granroth-Wilding and Clark, 2016) is a machine reading comprehension task designed to evaluate the model's ability to solve cloze problems. In the MCNC task, given the context of a story or narrative text, the model needs to select the most appropriate option from multiple alternatives to fill in the blanks in the text, making the whole story or text more coherent and complete.

4.3. Comparison methods

We compare our method with some baselines, and we briefly introduce these baseline methods below.

Predicate Tensor (Weber et al., 2018) and **Role-factor Tensor** (Weber et al., 2018) are models that use tensors to learn the interactions between the predicate and its arguments and are trained using co-occurring events as supervision. **SAM-Net** (Lv et al., 2019) tries to simulate the process that human beings tend to choose limited key information for memorizing and extracting answers selectively. **KGEB** (Ding et al., 2016) incorporates knowledge graph information. **FEEL** (Lee and Goldwasser, 2018) and **UniFA-S** (Zheng et al., 2020) adopt discourse relations. **NTN-IntSent** (Ding et al., 2019) takes intent and sentiment into event representation learning as external knowledge. **SWCC** (Gao et al., 2022) learns representations using weakly supervised contrastive learning and clustering algorithms. For downstream tasks, we compare the following methods. We do not compare supervised representation learning (Bai et al., 2021; Lv et al., 2020), because we believe that purer event representations are more valuable, and we tend to discover internal feature information from the events themselves. **Random** picks a candidate at random uniformly. **PPMI** (Chambers and Jurafsky, 2008) uses co-occurrence information and calculates Positive PMI for event pairs. **BiGram** (Jans et al., 2012) calculates bi-gram conditional probabilities based on event term frequencies. **Word2Vec** (Mikolov et al., 2013) uses the word embeddings trained by the Skipgram algorithm and event representations are the summation of word embeddings of predicates and arguments.

5. Experiment Results

Table 2 shows the performance of different model methods on the event similarity task. The results show that our proposed MCTM model offers the best performance among the comparison methods on two hard similarity tasks. And there is also a certain improvement in the task of transferring events. It outperforms the Role-factor Tensor method based on co-occurrence information and methods trained with additional annotations and commonsense knowledge, such as NTN-IntSent and UniFA-S. Compared with SWCC, which also uses the contrastive learning method, the performance of MCTM is also greatly improved, which proves the superiority of the multi-granularity labeling method. Table 3 reports the performance of different methods on the MCNC task. The table shows that MCTM achieves the best accuracy on the MCNC task under the zero-shot transfer setting, indicating that the proposed MCTM generalizes better to downstream tasks than other comparison methods.

6. Ablation Study

To explore the influence of different methods in the model on the model, we conduct an ablation experiment, as shown in Table 4. We remove a

Model	Accuracy (%)
Random	20.00
PPMI	30.52
BiGram	29.67
Word2Vec	37.39
SWCC	44.50
MCTM	**46.15**

Table 3: The performance on the MCNC task. The best results are bolded.

Model	**OHS (%)**	**EHS (%)**	**TSS(ρ)**
MCTM	**81.7**	**75.2**	**0.85**
Single Fine-grained	80.9 (↓0.8)	72.1 (↓3.1)	0.81 (↓0.04)
Single Coarse-grained	79.1 (↓2.6)	70.6 (↓4.6)	0.82 (↓0.03)
w/o MoE Layer	79.9 (↓1.8)	73.9 (↓1.3)	0.84 (↓0.01)

Table 4: Ablation experiments of different methods on the event similarity task. OHS: original hard similarity. EHS: extend hard similarity. TSS: transitive sentence similarity.

certain component in the model and examined the corresponding performance of the incomplete MCTM on the event similarity task. Since the multi-grained labeling shows the distinction between fine-grained and multi-granularized, we first test the performance of different scales of granularity separately, and the results show that the coarse-grained labeling method has a more significant impact on the extended event similarity task. Second, we test the MoE layer and showed that removing this module would produce a drop of around 1 point on the hard similarity task.

Next, we designed ablation experiments to discuss the impact of the sparsity of MoE on model performance. The sparsity of MoE in our model is reflected in the selection of experts. The sparse gating mechanism helps to reduce computational complexity so that only a few expert groups will be activated, thereby improving the efficiency of the model. For this part of the ablation experiment, we used several methods: 1). Close the gating mechanism: so that all experts are always active. 2). Reduce the sparsity of the gating mechanism: Gradually reduce the sparsity of the gating mechanism, that is, increase the probability of selecting multiple experts. 3). Randomly select expert group: At each time step or sample, experts are randomly selected to process the input data, instead of being selected according to our proposed top k mechanism. Table 5 presents the experimental results. We can see that sparsity has a greater impact on model performance, but has a smaller impact on the TSS task. We consider that it may be because the evaluation criteria of TSS are not similar.

We also conducted corresponding ablation experiments on Equation 21 and Equation 22. Here we consider separately: 1) Remove the $\mathcal{L}_I$ loss, that is, do not consider the same importance weight of each expert, which may cause the model to deviate when considering each granularity 2) Remove the $\mathcal{L}_L$ loss, that is, do not consider the model Load balancing. The following table shows our current experimental results. It can be seen in Table 6that the loss of these two parts has a greater impact on the performance of the model.

Model	**OHS (%)**	**EHS (%)**	**TSS(ρ)**
MCTM	**81.7**	**75.2**	**0.85**
Active all experts	80.2(↓1.5)	73.8(↓1.4)	0.84(↓0.01)
Multiple experts	79.9(↓1.8)	73.5(↓1.7)	0.84(↓0.01)
Random experts	80.1(↓1.6)	72.9(↓2.3)	0.83(↓0.02)

Table 5: Ablation experiment to verify the impact of MoE sparsity on model performance.

Model	**OHS (%)**	**EHS (%)**	**TSS(ρ)**
MCTM	**81.7**	**75.2**	**0.85**
w/o $\mathcal{L}_I$	76.7(↓5)	70.3(↓4.9)	0.79(↓0.06)
w/o $\mathcal{L}_L$	77.4(↓4.3)	72.2(↓3)	0.79(↓0.06)

Table 6: Ablation experiments to verify the impact of the losses in Equations 21 and 22 on model performance.

7. Analysis

In this section, we further analyze the fit of MoE and the idea of multi-granularity.

Number of experts. Figure 7 shows the overall performance of the model when we set different numbers of experts. We can only use no more than 15 experts due to the limitations of graphics card equipment. It can be concluded from the experiment that when the number of experts is less than 9, the overall performance is improved, but when the number of experts is more than 9, the performance of the model is improved little or even has a downward trend. We analyze that it may be because when too many experts are assigned, the calculation of weight distribution may lead to deviations in the calculation of features.

Importance weights at different granularities. We have generated importance weights at different granularities during the training process, as illustrated in a, b and c in Figure 4. Figure d in 4 shows a pie chart of the weight distribution for each granularity. We can see that the Pre-obj Labeling particle has the highest weight of the three grains (37%). The importance weight associated with the predicate-object labeling granularity exhibits a consistent upward trend throughout the training, eventually stabilizing at approximately 37%. In contrast, the weight for random labeling granularity consistently decreases and remains stable at 30%. The subject-predicate labeling weight, on the other hand, fluctuates between 33% and 33.4% overall.

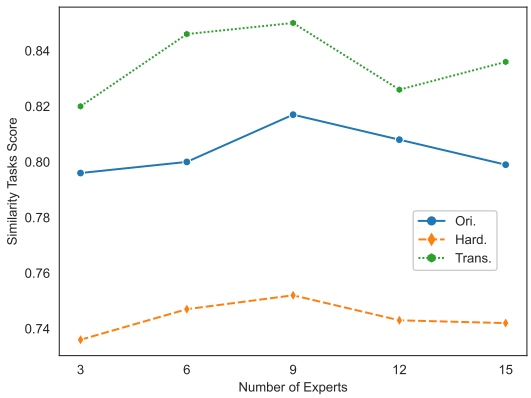
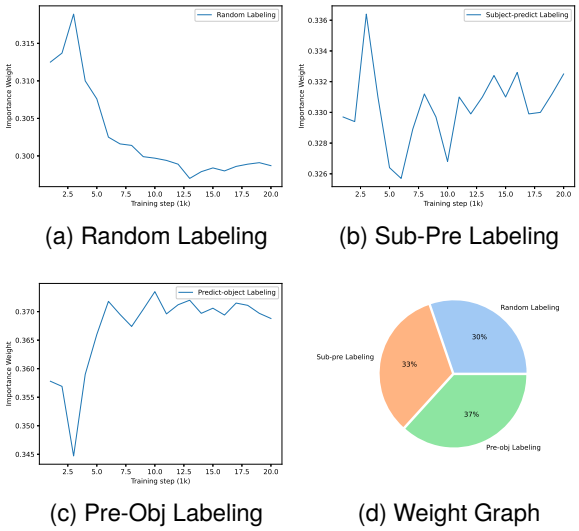


Figure 3: Effect of different numbers of experts on model performance.



(a) Random Labeling (b) Sub-Pre Labeling

(c) Pre-Obj Labeling (d) Weight Graph

Figure 4: Figures a, b and c show the changes in the importance weights of three different granularities during training. Figure d shows the weight ratio of the final three granularities.

Our analysis suggests that as the model learns the semantic features of simple events, there is a strong connection between the predicate and the object. Many events exhibit distributed similarities within these two grammatical structures. This observation could explain the increasing weight assigned to the predicate-object labeling granularity over time.

8. Related Work

Event representation learning. Event representation learning plays a crucial role in understanding the relationships between events (Wadden et al., 2019; Gao et al., 2019; Yu et al., 2020). For instance, the script event prediction task relies on event relationships (temporal, causal, and other complex relationships) to explore the connection between preceding events and predict the most likely subsequent event (Zhou et al., 2022; Ding et al., 2019; Li et al., 2018a). Furthermore, event schema induction (Li et al., 2020) and event narrative modeling (Li et al., 2018a; Xu et al., 2022; Lee and Goldwasser, 2019) are additional downstream tasks that are currently under investigation. The incorporation of external knowledge also enhances event feature representation. Examples include the utilization of external commonsense knowledge (Ding et al., 2019) as well as human action intentions and sentiments (Sap et al., 2019).

Contrastive learning. The method of contrastive learning has applications in many tasks in several fields (Khosla et al., 2020; Xiao et al., 2021; Chen and He, 2021). Building upon prior research in contrastive learning (Xiao et al., 2021; Zimmermann et al., 2021; **?**), we observe that negative examples hold greater significance than positive examples in the context of contrastive learning. Therefore, our work focuses on negative examples and incorporates multi-granularity labeling to facilitate learning in conjunction with positive examples. This approach is motivated by the achievements of multi-grained pre-trained language models (Joshi et al., 2020; Diao et al., 2020; Zhang et al., 2021), which have demonstrated success in a range of applications.

Mixture of Experts. Mixture of Experts (MoE) has witnessed significant advancements in recent years, with researchers introducing various modifications to its structure. The work by (Lepikhin et al., 2021) was the first to extend the concept of MoE to Transformer models. The specific approach involves replacing every other position-wise Feed-Forward Network (FFN) layer in both the encoder and decoder of the Transformer with an MoE layer. Switch Transformer (Fedus et al., 2021) introduces a gating network that routes to only one expert at a time, thus achieving higher computational efficiency for the MoE layer alone. Google's super-large model, introduced in 2021 (Du et al., 2022), surpasses GPT-3 (Brown et al., 2020) in performance on 29 NLP tasks, despite being three times larger. This accomplishment is attributed to the design of the Sparse MoE, which reduces the training cost to only one-third of GPT-3.

9. Conclusion

In our work, we introduce MCTM, a method that incorporates multi-granularity labeling of positive examples to enhance token-level knowledge in contrastive learning. Specifically, we utilize the Mixture

of Experts Layers to learn distinct parameters for different granularities, allowing for a more effective understanding of event characteristics. Experimental results demonstrate that our model outperforms other baselines in the event similarity task. Additionally, the results showcase the ability of our model to learn implicit relationships between different events through downstream tasks.

10. Acknowledgements

This work is supported by grant from the National Natural Science Foundation of China (No. 62076048), the Science and Technology Innovation Foundation of Dalian (2020JJ26GX035).

11. Bibliographical References

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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

Hong Chen, Raphael Shu, Hiroya Takamura, and Hideki Nakayama. 2021. GraphPlan: Story generation by planning with event graph. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 377–386, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758.

Shumin Deng, Ningyu Zhang, Luoqiu Li, Chen Hui, Tou Huaixiao, Mosha Chen, Fei Huang, and Huajun Chen. 2021. OntoED: Low-resource event detection with ontology embedding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2828–2839, Online. Association for Computational Linguistics.

Jacob Devlin, Mingwei 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.

Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. 2020. ZEN: Pre-training Chinese text encoder enhanced by n-gram representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4729–4740, Online. 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.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2016. Knowledge-driven event embedding for stock prediction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2133–2142, Osaka, Japan. The COLING 2016 Organizing Committee.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity.

Jun Gao, Wei Wang, Changlong Yu, Huan Zhao, Wilfred Ng, and Ruifeng Xu. 2022. Improving event representation via simultaneous weakly supervised contrastive learning and clustering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3036–3049, Dublin, Ireland. Association for Computational Linguistics.

Lei Gao, Prafulla Kumar Choubey, and Ruihong Huang. 2019. Modeling document-level causal structures for event causal relation identification. 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 1808–1817, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

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, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2727–2733. AAAI Press.

Zhiting Hu, Haoran Shi, Bowen Tan, Wentao Wang, Zichao Yang, Tiancheng Zhao, Junxian He, Lianhui Qin, Di Wang, Xuezhe Ma, Zhengzhong Liu, Xiaodan Liang, Wanrong Zhu, Devendra Sachan, and Eric Xing. 2019. Texar: A modularized, versatile, and extensible toolkit for text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 159–164, Florence, Italy. Association for Computational Linguistics.

Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6384–6392.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Bram Jans, Steven Bethard, Ivan Vulic, and Marie-Francine Moens. 2012. Skip n-grams and ranking functions for predicting script events. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, pages 336–344. ACL; East Stroudsburg, PA.

Michael I Jordan and Robert A Jacobs. 1994. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Yannis Kalantidis, Mert B Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard negative mixing for contrastive learning. *Advances in neural information processing systems*, 33:21798–21809.

Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. 2014. A study of entanglement in a categorical framework of natural language. *arXiv preprint arXiv:1405.2874*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

I-Ta Lee and Dan Goldwasser. 2018. Feel: Featured event embedding learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

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.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. Gshard: Scaling giant models with conditional computation and automatic sharding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare Voss. 2020. Connecting the dots: Event graph schema induction with path language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 684–695, Online. Association for Computational Linguistics.

Zhongyang Li, Xiao Ding, and Ting Liu. 2018a. Constructing narrative event evolutionary graph for script event prediction. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19,*

2018, Stockholm, Sweden, pages 4201–4207. ijcai.org.

Zhongyang Li, Xiao Ding, and Ting Liu. 2018b. Generating reasonable and diversified story ending using sequence to sequence model with adversarial training. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1033–1043, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Shangwen Lv, Wanhui Qian, Longtao Huang, Jizhong Han, and Songlin Hu. 2019. Sam-net: Integrating event-level and chain-level attentions to predict what happens next. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6802–6809. AAAI Press.

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

Lara J. Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark O. Riedl. 2018. Event representations for automated story generation with deep neural nets. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 868–875. AAAI Press.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Sumegh Roychowdhury, Sumedh A Sontakke, Laurent Itti, Mausoom Sarkar, Milan Aggarwal, Pinkesh Badjatiya, Nikaash Puri, and Balaji Krishnamurthy. 2022. Sherlock: Self-supervised hierarchical event representation learning. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 2672–2678. IEEE.

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. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3027–3035. AAAI Press.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V Le, Geoffrey E Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: enhanced representation through knowledge integration. *ArXiv preprint*, abs/1904.09223.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A continual pre-training framework for language understanding. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020,*

The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 8968–8975. AAAI Press.

Prashanth Vijayaraghavan and Deb Roy. 2021. Lifelong knowledge-enriched social event representation learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3624–3635, Online. Association for Computational Linguistics.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. 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 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Junjie Wang, Yuxiang Zhang, Lin Zhang, Ping Yang, Xinyu Gao, Ziwei Wu, Xiaoqun Dong, Junqing He, Jianheng Zhuo, Qi Yang, et al. 2022. Fengshenbang 1.0: Being the foundation of chinese cognitive intelligence. *ArXiv preprint*, abs/2209.02970.

Noah Weber, Niranjan Balasubramanian, and Nathanael Chambers. 2018. Event representations with tensor-based compositions. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4946–4953. AAAI Press.

Tete Xiao, Xiaolong Wang, Alexei A Efros, and Trevor Darrell. 2021. What should not be contrastive in contrastive learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Guangxuan Xu, Paulina T Isaza, Moshi Li, Akintoye Oloko, Bingsheng Yao, Aminat Adebeyi, Yufang Hou, Nanyun Peng, and Dakuo Wang. 2022. Nece: Narrative event chain extraction toolkit. *ArXiv preprint*, abs/2208.08063.

Fuzhao Xue, Ziji Shi, Futao Wei, Yuxuan Lou, Yong Liu, and Yang You. 2022. Go wider instead of deeper. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8779–8787.

Changlong Yu, Hongming Zhang, Yangqiu Song, Wilfred Ng, and Lifeng Shang. 2020. Enriching large-scale eventuality knowledge graph with entailment relations. *ArXiv preprint*, abs/2006.11824.

Xinsong Zhang, Pengshuai Li, and Hang Li. 2021. AMBERT: A pre-trained language model with multi-grained tokenization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 421–435, Online. Association for Computational Linguistics.

Jianming Zheng, Fei Cai, and Honghui Chen. 2020. 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 2020, Virtual Event, China, July 25-30, 2020*, pages 249–258. ACM.

Pengpeng Zhou, Bin Wu, Caiyong Wang, Hao Peng, Juwei Yue, and Song Xiao. 2022. What happens next? combining enhanced multilevel script learning and dual fusion strategies for script event prediction. *International Journal of Intelligent Systems*, 37(11):10001–10040.

Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. 2021. Contrastive learning inverts the data generating process. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12979–12990. PMLR.