



Generative Event Extraction via Internal Knowledge-Enhanced Prompt Learning

Hetian Song^{1,2}, Qingmeng Zhu², Zhipeng Yu², Jian Liang², and Hao He²(✉)

¹ University of Chinese Academy of Sciences, Beijing, China
songhetian20@mailsucas.ac.cn

² Institute of Software, Chinese Academy of Sciences, Beijing, China
{hetian2020, qingmeng, yuzhipeng, liangjian, hehao21}@iscas.ac.cn

Abstract. Event extraction is a crucial research task in information extraction. In order to maximize the performances of the pre-trained language model (PLM), some works formulating event extraction as a conditional generation problem. However, most existing generative methods ignore the prior information between event entities, and are usually over-dependent on hand-crafted designed templates, which causing subjective intervention. In this paper, we propose a generative event extraction model named KEPGEE based on internal knowledge-enhanced prompt learning. We firstly use relational graph neural networks (RGCN) to encode the event triples entities and fuse them with the word embeddings to obtain the knowledge representation. Then the knowledge representation is concatenated with task-specific virtual tokens to compose knowledge-enhanced soft prompts, which can provide additional event information to adapt the sequence-to-sequence PLM for the generative event extraction task. Besides, in template design, we add the related topic words into the prompt templates to enhance the implicit event information. We evaluate our model on ACE2005 and ERE datasets, and the results show that our model achieves matched or better performances with several classification-based or generation-based event extraction models (including the state-of-the-art models).

Keywords: Event Extraction · Generative · Prompt · Knowledge

1 Introduction

Event extraction (EE) is a challenging task for natural language understanding and cognition in context, which aims to extract events for structured data from a piece of contexts. Each event consists of triggers and arguments with their specific event roles [6].

The general event extraction work considers the identification of event triggers and event arguments as classification-based tasks, including in a pipeline

H. Song and Q. Zhu—Contributed equally to this work. This work is supported by the National Key R&D Program of China (2022YFC3103800) and National Natural Science Foundation of China (62101552).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
L. Iliadis et al. (Eds.): ICANN 2023, LNCS 14258, pp. 90–102, 2023.
https://doi.org/10.1007/978-3-031-44192-9_8

paradigm that models the two sub-targets separately [3, 7, 26], or in a joint formulation which constructs an end-to-end model to extract triggers and arguments simultaneously [1, 13, 23].

Recently, to maximize the performances of the pre-trained language model, some works formulating event extraction as a conditional generation problem. These methods usually make a PLM output the conditional generation sequences by hand-crafted designed template [8, 15]. In comparison with the classification-based methods, generative modeling methods could be more effective for low-resource, achieving competitive performances without complex structural modifications.

However, generation-based event extraction methods encounter two momentous obstacles to better performance. The one is **static event information**: Recent generation-based methods attempt to learn the event information by template, regardless of the prior information between event entities. The other is **over-dependence of predefined template**: The performances of generative methods are usually over-dependent on hand-crafted designed template, which is lack of internal information of the corresponding events [8, 15].

In this paper, to address above challenges, we propose KEPGEE (**K**nowledge-**E**nhanced **P**rompt-based **G**enerative **E**vent **E**xtraction), a generation-based event extraction model with knowledge-enhanced soft prompts. Specifically, to capture the internal event knowledge into generative PLM, we firstly use relational graph neural networks (RGCN) to encode the event triples entities from given resources. Secondly, we design a semantic fusion module to align semantics association between words and event entities, using predefined prompt templates for the former and knowledge graphs for the latter. Then the fused knowledge representation (latent embeddings) is concatenated with task-specific virtual tokens (trainable embeddings) to compose knowledge-enhanced soft prompts. Finally we add them into the encoder of BART [10] to adapt the sequence-to-sequence PLM for generation-based event extraction task. Furthermore, following the prompt template design of previous works [8, 15], we introduce a variant of Variational Auto-Encoders (VAE) model to get the topic words of event sentences as additional event priori information, and add them into the prompt templates. Our prompt templates leverages the implicit event knowledge effectively for conditional generation, and can be seen as an end-to-end method that solves event detection and event argument extraction simultaneously.

Contributions. The main contributions are as follows:

- (1) We propose a novel generative event extraction model named KEPGEE, which is based on internal knowledge-enhanced prompt learning. This method is able to provide additional internal event entity information.
- (2) Based on the original prompt templates, we add the related topic words into the prompt templates, and the topic words are generated by a neural topic model. This is an effective way to enhance the implicit event relating.
- (3) We evaluate our model KEPGEE on ACE2005 and ERE datasets. The experimental results show that our model achieves matched or competitive performances with several classification-based model or generation-based

model for event extraction (including the state-of-the-art models). KEPGEE also performs well in low-resource settings.

2 Related Work

2.1 Classification-Based Event Extraction

Event extraction is an important task in the field of information extraction, and has been studied for a long time [2,6]. Traditionally, related works follow the sequence labeling classification modeling approach. DMCNN [3] is a classical model for event extraction by using two dynamic multi-pooling convolutional neural networks to classify trigger and argument words. PLMEE [26] is implemented for trigger extraction and argument extraction via double BERT [4]. Lin et al. [13] propose OneIE, which incorporates global features and employs beam search. Additionally, some works formulate EE as a machine reading comprehension problem, which constructs questions and query model to get triggers and arguments [7].

2.2 Generation-Based Event Extraction

There have recently been some works that formulate event extraction as a conditional generation problem. Lu et al. [18] encode the input context to a tree-like event structure, and parse the generation sentences to get corresponding structured events. Li et al. [12] attempt to mark the trigger words and design the event templates by using <arg> as a placeholder for argument extraction. DEGREE [8] is used for low-resource event extraction by predefined prompt templates, which incorporates explicit event knowledge. GTEE-DYNPREF [15] integrates context information via event-specific prefixes to establish links between different event type.

2.3 Prompt Tuning

Prompt-tuning is a new paradigm for adapting pre-trained language models, and has achieved outstanding performances in several downstream tasks with the help of textual prompts [14]. The method of designing hand-crafted prompt templates is called “hard prompts”, which depends on mapping from class labels to answer tokens [17], and this method is effective in low-resource settings.

There are also some methods that optimize a series of embeddings into transformer, and can be seen as “soft prompts” [9,16], which focus on utilizing an abstract vector as the prompt template rather than label words. Besides, several works attempted to improve the performance of soft prompts by pre-training [22] or incorporating external knowledge [22,24].

2.4 Neural Topic Model

Topic model is a successful text analysis technology to mine the internal topics in the corpus, which is based on statistical methods. In recent years, variation auto-encoder (VAE) structure is the widely used in neural topic model (NTM), such as [19]. Additionally, Dieng et al. [5] propose an embedded topic model (ETM), which is a generative model of documents with word embeddings. [25] extract topic words by semantic correlation graphs. Recently, Li et al. [11] attempt to use a contrastive learning framework for topic mining.

3 Methods

In this section, we introduce our generative event extraction method KEPGEE, and the composition of prompt templates design. The model framework is shown in Fig. 1.

3.1 Overview of the Approach

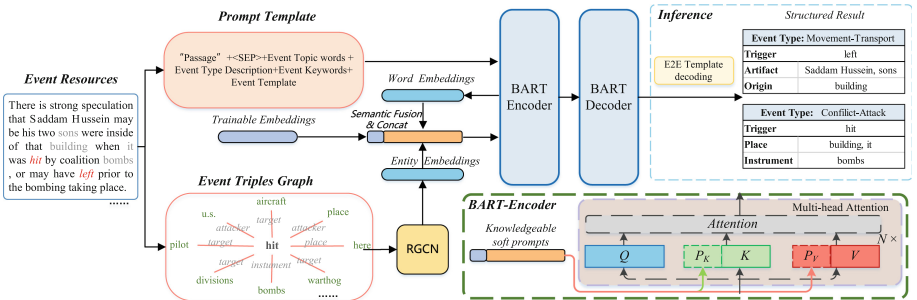


Fig. 1. The main structure of KEPGEE. We first encode known event triples entities by RGCN and generate the corresponding event prompt templates, then use a fusion module to align word-entity embeddings and concatenate with a series of trainable embeddings to construct soft prompts. Secondly, we integrate soft prompts into BART-Decoder. Finally, we decode the generative sentences to obtain event structure results.

Problem Statement. We conduct event extraction as a conditional generation task. We assume that we are given the event data sources \mathcal{D} with an event type set $\varepsilon = \{e_i \mid i \in [1, |\varepsilon|]\}$. The inputs D_{gen} sent to model for event type e_i consists of context \mathcal{C} and a specific predefined prompt template S_{e_i} . The generative output is A_{e_i} , which contains the event records in the original placeholder position.

The PLM. In our method, we take a pre-trained encoder-decoder language model BART [10] as our basic architecture. The text generation process models the conditional probability of selecting a new token given the previous tokens and the input to the encoder.

3.2 Training and Inference

Training. The Training objective of our model KEPGEE is to generate an output which is similar to event prompt templates. In this way, the position of original placeholders are replaced by generated words. As shown in Fig. 1, <Trigger> is expected to be replaced by the trigger word “left”, and some-place is expected to be replaced by the argument word “building” for role “Origin”, something is replaced by “Saddam Hussein and sons” for the same role “Artifact”. Specially, we assume that ϕ is trainable parameters. For event extraction task, we use prompt-augmented context D_{gen} which is generated in training set \mathcal{D} to derive the prediction loss for learning ϕ , which is formally given as:

$$\begin{aligned} \mathcal{L}_\phi(\mathcal{D}) &= - \sum_{j=1}^{|\mathcal{D}|} \sum_{i=1}^{|\varepsilon|} \log p(G_{e_i, C_j} | X_{e_i, C_j}) \\ X_{e_i, C_j} &= [P_k; D_{gen, C_j}] \\ D_{gen, C_j} &= [C_j; [SEP]; S_{e_i}] \end{aligned} \tag{1}$$

where C_j is j-th context in event sources \mathcal{D} . G_{e_i, C_j} is j-th ground truth sequence by filling the gold words of event records replacing the placeholders, and D_{gen, C_j} is j-th prompt-augmented context. P_{k_j} indicates j-th soft prompts.

Inference. Similarly with [8], we generate a corresponding output by enumerating all event types. After that, we compare the outputs with the predefined event template and apply slot mapping to determine the predicted triggers and arguments. We choose the closest one to the trigger span for argument predictions. We set an acceptable sequence length and make our model generate the sequence by $BEAM = 4$.

3.3 Soft Prompts Components

Even if existing generative methods have achieved competitive performance for event extraction, they lack the internal prior information between event entities. Following previous works [24], we attempt to incorporate event-triples-KG from given event data sources \mathcal{D} , since it provides prior knowledge about event entities.

Knowledge Encoding. We first encode the entities to event-triples-KG. We use relational graph neural networks (RGCN) [20] to obtain the event entity embeddings, which can construct the event relational semantics by information aggregation and flow. The derived entity matrix is $E = [h_1^E, h_2^E, \dots, h_{n_e}^E]$,

where n_e is the number of event entities. Specially, we regard the output embeddings of BART-Encoder as word embeddings, which encode the prompt-augmented sources \mathcal{D}_{gen} to learn the particular format and information of corresponding event context. Similarly, the word embeddings matrix is denoted as $T = [h_1^T, h_2^T, \dots, h_{n_w}^T]$, n_w is the length of input sequence.

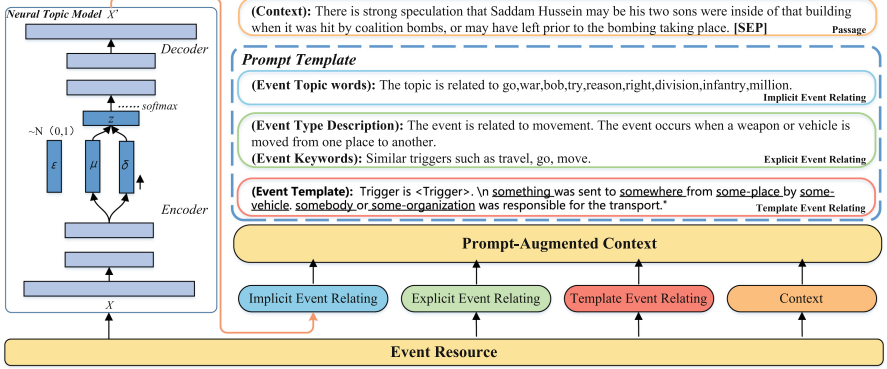


Fig. 2. The example of a prompt template for *MOVEMENT:TRANSPORT* event.

Word-Entity Alignment. To align the semantic representation between words and entities, we use a cross interaction algorithm to associate with these two embeddings:

$$\begin{aligned} M &= T^\top W E \\ T' &= T + E M \end{aligned} \quad (2)$$

where M is the correlation matrix between the two embeddings and T' denotes the fused word-entity representations. W is the transformation matrix. We establish the semantic association between words and entities via the simple transformation above.

The Soft Prompts Design. Specially, we concatenate the word-entity embeddings T' with task-specific soft prompts P_{ori} to construct the knowledge-enhanced soft prompts P_k . The task-specific soft prompts P_{ori} are series of trainable embeddings and can be considered as if they were virtual tokens. They are usually re-parameterized by a feed-forward network, which consists of two linear transformations with a *TANH* activation function in between. The formal description of P_k is:

$$P_k = [P_{ori}; T'] \quad (3)$$

To limit the format of generating sequences in better, we then transfer P_k and concatenate it with the key-value pairs K and V respectively of the BART-Encoder attention layers rather than the BART-Decoder.

3.4 Prompt Template Design

The event templates we use are based on the design of [8], and we add the topic words of corresponding event sentences via a neural topic model. Our prompt template design method can relieve the problem of over-dependence, and make the model learn the implicit event information.

The Components of Prompt Templates. An example of the prompt template is shown in Fig. 2. We concatenate each prompt template with corresponding original context as prompt-augmented sources \mathcal{D}_{gen} , which are generated from the given event data sources \mathcal{D} . We acquire the generative output sequences A_{e_i} , which contain the event records in the original placeholder position. Every prompt template contains the following components: the *event topic words* which are extracted by a neural topic model, the *event type description*, the *event keywords*, and the *event template*. Specially, we regard the topic words as **implicit event relating**, because these topic words may not mentioned in the corresponding event context, but in the given event data sources \mathcal{D} . Similarity, we regard the event type description and event keywords as **explicit event relating**, for these elements could be acquired directly. We also regard the event template as **template event relating**, which guide the model to generate triggers and arguments at the locations of placeholders.

Implicit Event Relating. We introduce the topic words in prompt templates for providing related event information about corresponding context, because the hand-crafted designed prompts may cause the subjective intervention. We use a variant of classical topic model W-LDA [19] to obtain the topic words. The model is composed of an encoder and a decoder, working resembling the data reconstruction process. We assume that V is the vocabulary of given event data sources \mathcal{D} , and j -th context is represented as $C_j = (x_1, x_2, \dots, x_n)$, where n is the length of this context. The expected topic type num is K which is predefined, and each topic $k = 1, \dots, K$ is a probability distribution over the words in the vocabulary V . We also assume that every context has its related topics $\theta \in \mathbb{R}^K, \sum_k \theta_k = 1, \theta_k \geq 0$. The flow path for getting topic words is as follows. We First get the intermediate embeddings μ and $\log \sigma$ via the encoder:

$$\mu = f_\mu(C_j) \tag{4}$$

$$\log \sigma = f_\sigma(C_j) \tag{5}$$

where μ and σ are the hyper-parameters of sampling topic-word distribution, and f_μ and f_σ are the two feed-forward networks which have the same structure. We use Gaussian distribution to generate topic words, and the the topic implicit variable of given context z is also subject to multi-dimensional Gaussian distribution.

$$z \sim N(\mu, \sigma^2) \quad (6)$$

$$\theta = \text{softmax}(z) \quad (7)$$

$$p_w = \text{softmax}(W_\varphi \theta) \quad (8)$$

where $W_\varphi \in R^{K \times V}$ represents the distribution matrix of topic words. We take given event data sources \mathcal{D} as training data to reconstruct the predicted word probability. The model takes the Gaussian mixture distribution as priori distribution. We use above topic model to obtain the topic words of each context. Specially, we set the expected topic type num $K = 30$. We choose Top3 related topics and select Top3 words for each topic, amount of 9 words to add into the prompt template of corresponding context as implicit event relating. The topic words prompt is shown in Fig. 2.

Explicit Event Relating. We regard the event type description and event keywords as explicit event relating, because they are clearly prompted by the given event context. For each event type, we construct a unique event description which is annotated from given event data sources \mathcal{D} , and we treat some trigger words that are semantically related to the given event type as keywords.

Template Event Relating. We take the event template as template event relating, which defines the output format and slots for predicting. Firstly, we hope to detect the trigger word via replacing the placeholder “< Trigger >” of “Trigger is < Trigger >”. Secondly, we hope to obtain the argument words related to events. In event template, we use some placeholders which are starting with “some” to represent argument roles, and make the model generate target arguments replacing these placeholders. Every event type has its own unique event template.

4 Experiments

In this section, we evaluate the performances of our model KEPGEE by conducting experiments on ACE2005 and ERE datasets.

4.1 Experiment Setup

Dataset. We conduct our experiments on two widely used datasets, ACE2005 [6] and ERE [21]. The former contains 33 event types and 22 argument roles, the latter contains 38 event types and 21 argument roles. Specifically, we choose English part and adopt the pre-process method in [13].

Experimental Details. We use the HuggingFace implementation of the pre-trained encoder-decoder language model BART [10]. We set default prompt length to 20, input length to 250 on ACE2005 and 375 on ERE, max output length to 130. Epoch is 30, batchsize is 16 and the learning rate is set to 1e-5. We report average performance and the best methods are bold.

Evaluation Metrics. We use the same criteria in previous works [8, 15]. We report the F1-scores of trigger classification (Trg-C) and argument classification (Arg-C) in major. We also report F1-scores of trigger identification (Trg-I) and argument identification (Arg-I) in sub-experiments. **Trg-I**: an trigger is identified correctly if its offset matches the ground truth. **Trg-C**: an trigger is classified correctly if its offset and event type both match the ground truth. **Arg-I**: an argument is identified correctly if its offset and event type both match the ground truth. **Arg-C**: an argument is classified correctly if its offset, event type and role all match the ground truth.

Compared Baselines. We compare our method KEPGEE with following classification-based models and generation-based models. **DYGIE++** [23]: a span-based method which introduces a graph structure to capture the interaction of span. **BERT_QA** [7]: a MRC-based method which uses QA pairs to classify the position of the predicted span. **OneIE** [13]: a joint-based method which employs global features to make decisions between instances and sub-tasks. **TEXT2EVENT** [18]: a generation-based method which converts the input sequences to the tree-like structure and generates target words. **DEGREE** [8]: a conditional generation method which uses prompt templates to obtain the triggers and arguments. **GTEE-DYNPREF** [15]: a conditional generation method which is also enhanced by static and dynamic prefixes.

4.2 Main Result

The main results on ACE2005 and ERE are shown in Table 1. We display the F1-scores of trigger classification (Trg-C) and argument classification (Arg-C). We take methods into two groups, the one is group of classification-based models, and the other is group of generation-based models. We implement our model in BART-large, which is similar with other generation-based methods. The model with * represents that the numbers are from previous paper.

Table 1. The Main results on ACE2005 and ERE datasets.

Model	Type	ACE2005-EN		ERE-EN	
		Trg-C	Arg-C	Trg-C	Arg-C
DyGIE++*	Cls	69.7	48.8	–	–
BERT_QA*	Cls	72.3	53.3	–	–
OneIE*	Cls	72.8	54.8	57.0	46.5
Text2Event*	Gen	71.8	54.4	59.1	50.5
DEGREE	Gen	73.5	55.3	57.2	49.8
GTEE-DYNPREF*	Gen	74.3	54.7	66.9	55.1
KEPGEE(ours)	Gen	76.2	58.3	64.9	54.6

Our proposed model KEPGEE achieves great performances for Trg-C and Arg-C compared with other baselines. KEPGEE outperforms OneIE, which is the state-of-the-art classification-based model, by 3.4/3.5% increase on ACE2005 and 7.4/8.1% increase on ERE for Trg-C/Arg-C. Although the test results of KEPGEE are not better than the SOTA generation-based method GTEE-DYNPREF on ERE, our model is competitive enough and outperforms the others. Compared with GTEE-DYNPREF, we attempt to use soft prompts in a knowledge-enhanced manner rather than aggregating information from various event types. And compared with DEGREE, we introduce the topic words in corresponding templates via a neural topic model instead of completely hand-crafted prompt templates.

4.3 Result for Low-Resource

We conduct a experiment for low-resource event extraction. We re-implement DEGREE and our model in BART-large. Following the pre-process method in [8], we split training data into different proportions (1%, 3%, 5%, 10%, 20%, 30%, 100%) and use the original test set.

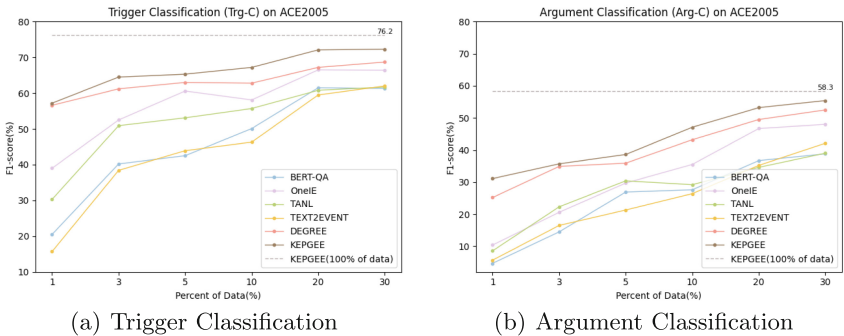


Fig. 3. The F1-scores for low-resource event extraction.

As visualized in Fig. 3, our model KEPGEE outperforms DEGREE (the SOTA model for low-resource) and other models both trigger classification and argument classification in general. This benefits from the introduction of internal knowledge. Specifically, we discover that KEPGEE could improve more in event argument-classification with more data. This demonstrates the effectiveness of internal knowledge-enhanced soft prompts, making the model learn to recognize argument words more effectively by entity knowledge.

4.4 Ablation Study

We conduct an ablation study for the components of KEPGEE. We implement our model in BART-large and train in ACE2005. Table 2 demonstrates how different components of KEPGEE affect the performance.

Table 2. The ablation study for the components of KEPGEE.

Model	ACE2005-EN			
	Trg-I	Trg-C	Arg-I	Arg-C
Full KEPGEE	80.9	76.2	60.8	58.3
- w/o word-entity knowledge	80.2	75.8	60.4	57.8
- only task-specific soft prompts	79.1	75.5	60.2	57.6
- only topic words	77.5	74.1	59.7	56.4
- w/o extra components (original DEGREE)	76.8	73.5	58.3	55.3

We simply consider that there are three optimized components than DEGREE: word-entity knowledge, task-specific soft prompts and topic words. We discover that introducing topic words into the prompt templates is effective for event argument extraction particularly, this could demonstrate that the topic words contain the implicit event relating information. Task-specific soft prompts play an important role, which could greatly improve the performance via contributing the virtual tokens in the attention layers. The word-entity knowledge is also important to event extraction task, because it makes the event entities to be highlighted in the corresponding context.

5 Conclusion

In this paper, we propose a generative event extraction model named KEPGEE, which is based on internal knowledge-enhanced prompt learning. Specifically, we firstly encode the event triples entities via relational graph neural networks and fuse them with the word embeddings to obtain the knowledge representation, then the knowledge representation is concatenated with task-specific virtual tokens to compose knowledge-enhanced soft prompts. Besides, we add the topic words into corresponding prompt templates to enhance the implicit event information. Moreover, our experimental results show that KEPGEE achieves matched or better performances with several state-of-the-art classification-based or generation-based event extraction models.

References

1. Abdelaziz, I., Ravishankar, S., Kapanipathi, P., Roukos, S., Gray, A.G.: A semantic parsing and reasoning-based approach to knowledge base question answering. In: AAAI, pp. 15985–15987 (2021)
2. Ahn, D.: The stages of event extraction. In: Proceedings of the Workshop on Annotating and Reasoning about Time and Events, pp. 1–8 (2006)
3. Chen, Y., Xu, L., Liu, K., Zeng, D., Zhao, J.: Event extraction via dynamic multi-pooling convolutional neural networks. In: ACL, pp. 167–176 (2015)

4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL, pp. 4171–4186 (2019)
5. Dieng, A.B., Ruiz, F.J., Blei, D.M.: Topic modeling in embedding spaces. *Trans. Assoc. Comput. Linguist.* **8**, 439–453 (2020)
6. Doddington, G.R., Mitchell, A., Przybocki, M.A., Ramshaw, L.A., Strassel, S.M., Weischedel, R.M.: The automatic content extraction (ACE) program - tasks, data, and evaluation. In: LREC (2004)
7. Du, X., Cardie, C.: Event extraction by answering (almost) natural questions. In: EMNLP, pp. 671–683 (2020)
8. Hsu, I., et al.: DEGREE: A data-efficient generation-based event extraction model. In: NAACL, pp. 1890–1908 (2022)
9. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: EMNLP, pp. 3045–3059 (2021)
10. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: ACL, pp. 7871–7880 (2020)
11. Li, J., Shang, J., McAuley, J.J.: Uctopic: Unsupervised contrastive learning for phrase representations and topic mining. In: ACL, pp. 6159–6169 (2022)
12. Li, S., Ji, H., Han, J.: Document-level event argument extraction by conditional generation. In: NAACL-HLT, pp. 894–908 (2021)
13. Lin, Y., Ji, H., Huang, F., Wu, L.: A joint neural model for information extraction with global features. In: ACL, pp. 7999–8009 (2020)
14. Liu, P., et al.: Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **55**(9), 1–35 (2023)
15. Liu, X., Huang, H., Shi, G., Wang, B.: Dynamic prefix-tuning for generative template-based event extraction. In: ACL, pp. 5216–5228 (2022)
16. Liu, X., et al.: P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In: ACL, pp. 61–68 (2022)
17. Lu, Y., Bartolo, M., Moore, A., Riedel, S., Stenetorp, P.: Fantastically ordered prompts and where to find them: overcoming few-shot prompt order sensitivity. In: ACL, pp. 8086–8098 (2022)
18. Lu, Y., et al.: Text2event: Controllable sequence-to-structure generation for end-to-end event extraction. In: ACL/IJCNLP, pp. 2795–2806 (2021)
19. Nan, F., Ding, R., Nallapati, R., Xiang, B.: Topic modeling with wasserstein autoencoders. In: ACL, pp. 6345–6381 (2019)
20. Schlichtkrull, M.S., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: ESWC. vol. 10843, pp. 593–607 (2018)
21. Song, Z., et al.: From light to rich ERE: annotation of entities, relations, and events. In: HLP-NAACL, pp. 89–98 (2015)
22. Vu, T., Lester, B., Constant, N., Al-Rfou, R., Cer, D.: Spot: Better frozen model adaptation through soft prompt transfer. In: ACL, pp. 5039–5059 (2022)
23. Wadden, D., Wennberg, U., Luan, Y., Hajishirzi, H.: Entity, relation, and event extraction with contextualized span representations. In: EMNLP-IJCNLP, pp. 5783–5788 (2019)
24. Wang, X., Zhou, K., Wen, J., Zhao, W.X.: Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In: KDD, pp. 1929–1937 (2022)

25. Wang, Y., Li, X., Zhou, X., Ouyang, J.: Extracting topics with simultaneous word co-occurrence and semantic correlation graphs: neural topic modeling for short texts. In: EMNLP, pp. 18–27 (2021)
26. Yang, S., Feng, D., Qiao, L., Kan, Z., Li, D.: Exploring pre-trained language models for event extraction and generation. In: ACL, pp. 5284–5294 (2019)