Contextualized Soft Prompts for Extraction of Event Arguments

Chien Van Nguyen, Hieu Man, and Thien Huu Nguyen
[1]Department of Computer Science, University of Oregon, Eugene, OR, USA
chienn@uoregon.edu, {hieum, thien}@cs.uoregon.edu

Abstract

Event argument extraction (EAE) is a subtask of event extraction where the goal is to identify roles of entity mentions for events in text. The current state-of-the-art approaches for this problem explore prompt-based methods to prompt pre-trained language models for arguments over input context. However, existing prompt-based methods mainly rely on discrete and manually-designed prompts that cannot exploit specific context for each example to improve customization for optimal performance. In addition, the discrete nature of current prompts prevents the incorporation of relevant context from multiple external documents to enrich prompts for EAE. To this end, we propose a novel prompt-based method for EAE that introduces soft prompts to facilitate the encoding of individual example context and multiple relevant documents to boost EAE. We extensively evaluate the proposed method on benchmark datasets for EAE to demonstrate its benefits with state-of-the-art performance.

1 Introduction

As an important task in Information Extraction (IE), Event Argument Extraction (EAE) aims to recognize event arguments and roles for given event mentions in text. For example, in the text "*On the morning of 1 March 2019, Taliban gunmen and suicide bombers **attacked** Camp Shorabak.*" with the event trigger "*attacked*" of type *Conflict.Attack*, the goal of EAE systems is to identify "*gunmen*" and "*bombers* as the *Attacker* argument, and "*Camp Shorabak*" as the *Target*. Along with event detection, EAE has important applications for different natural language processing (NLP) tasks.

EAE research progress has been accelerated by deep learning architectures to significantly boost extraction performance. Early deep learning models for EAE have followed the traditional approaches to formulate EAE as classification or sequence labeling problems (Chen et al., 2015; Nguyen et al., 2016; Sha et al., 2018; Liu et al., 2018; Nguyen and Nguyen, 2018; Pouran Ben Veyseh et al., 2022a). Recently, there has been a growing interest in solving EAE in the new question answering or text generation frameworks to better exploit task-specific information (e.g., labels/descriptions of argument roles) via prompts for pre-trained language models (PLM). As such, question answering methods for EAE create a question for each argument role to perform span extraction over input context (Du and Cardie, 2020; Liu et al., 2020; Li et al., 2020; Liu et al., 2021a) while text generation models directly consume an input text and argument-specified prompt/template to generate arguments for each event mention (Li et al., 2021; Zeng et al., 2022). However, a common issue in current prompt-based methods for EAE involves the use of discrete and manually-designed prompts to present task information for the models, e.g., event types and argument roles. As such, these prompts often follow some pre-defined templates (Li et al., 2021; Ma et al., 2022) that are applied to extract arguments for all events in text. While convenient for human understanding, discrete and pre-defined prompts might not be ideal for all examples, causing sub-optimal performance (Liu et al., 2021b). The discrete nature also makes it challenging to achieve prompt customization for each example in EAE models. Further, due to the employment of PLMs, it has been observed that model performance can be very sensitive to specific formulations of discrete prompts (Zhao et al., 2021; Ma et al., 2022), leading to instability and less reliability when adapting to different datasets.

Another issue of hard prompts for EAE models concerns other relevant examples from training data that can provide helpful information to support argument prediction for current input text and event type. As such, a few recent work has retrieved related examples for an input text to combine with hard prompts to improve EAE (Du et al., 2022; Du

Findings of the Association for Computational Linguistics: ACL 2023, pages 4352–4361
July 9-14, 2023

and Ji, 2022). However, due to the input length limit of PLMs, the number of relevant examples in the prompts is also constrained, thus unable to fully leverage their advantages to boost performance.

To this end, our work proposes a novel prompt-based method for EAE where learnable soft prompts are explicitly introduced to enable prompt customization for examples, stability improvement, and incorporation of relevant example context. In particular, based on the architecture of generative PLMs, our model directly utilizes input example representations to compute soft prompts for EAE, thus allowing the prompts to be specifically designed for each example for better customization. In addition, soft prompts facilitate the accumulation of representations of relevant examples for an input event type to consume more examples for richer prompts. To exploit this flexibility of soft prompts, our model extensively considers relevant examples as the texts in training data that contain similar event types to an input text, leading to comprehensive external event context to aid EAE. Accordingly, we introduce a graph structure to capture mentioning relations between documents and event types. This graph is then fed into graph neural networks to facilitate representation aggregation of relevant documents with similar events for soft prompt computation. We evaluate the proposed model for EAE on the benchmark datasets RAMS and WIKIEVENTS. The results demonstrate the benefits of the proposed method, leading to state-of-the-art performance for EAE.

2 Model

In EAE, given an event trigger/mention in a document, we need to identify argument spans and roles for the event. For convenience, let $\mathcal{D} = \{D_1, \ldots, D_{|\mathcal{D}|}\}$ be the set of documents in the training data and $e_k \in D_i$ be the current event trigger in document D_i with event type et for EAE.

Relevant Context Aggregation: Our EAE model follows the prompt-based framework (Ma et al., 2022) where a prompt is created for each event type and fed into the pre-trained language model BART (Lewis et al., 2020) to perform span extraction for the argument roles. As such, in contrast to hard and manually-designed prompts as in previous work, our model introduces soft prompts with example customization and relevant example aggregation to boost the performance. In particular, given the current event trigger $e_k \in D_i$, our model first aims to aggregate context representations from relevant documents in $\mathcal{D}$ for the event type et of e_k to enrich soft prompt computation. Motivated by relevant documents via similar event types, we we first construct an event-type mentioning graph $\mathcal{G}$ between the documents in $\mathcal{D}$ and the event types $\mathcal{T} = \{t_1, \ldots, t_{|T|}\}$ to facilitate representation aggregation. In particular, the node set $\mathcal{V}$ for our graph involves both documents and event types, i.e., $\mathcal{V} = \mathcal{D} \cup \mathcal{T}$. We only connect a document $D_u \in \mathcal{D}$ and an event type t_v in $\mathcal{G}$ if there exists an event mention of type t_v in D_u. In this way, we can link the documents in $\mathcal{D}$ via similar event type mentioning for convenient representation aggregation with graph neural networks.

To obtain representations for each document $D_u \in \mathcal{D}$, we introduce the markers <ET> and </ET> before and after each event trigger word in D_u to generate the marker-augmented document $\hat{D}_u$. The augmented document is then sent into the encoder of BART to produce a representation for each word in $\hat{D}_u$ (using the averages of hidden vectors in the last layer for sub-words). Afterward, the representation $\overline{D}_u^0$ for $D_u \in \mathcal{D}$ is computed by performing mean pooling over the representations for the <ET> markers of event triggers, aiming to retain event-focused context in the representation. For the event types $t_v \in \mathcal{T}$, we initialize their representations $\bar{t}_v^0$ randomly. Afterward, the graph $\mathcal{G}$ and representations $\overline{D}_u^0$ and $\bar{t}_v^0$ for documents and event types are consumed by a graph attention network (Veličković et al., 2017) to aggregate the representations via the connections in $\mathcal{G}$, producing richer representations $\overline{D}_u^L$ and $\bar{t}_v^L$ for D_u and t_v after L layers of transformation. Consequently, we treat the induced representation $\overline{et}$ for the current event type et as the aggregation for context information of relevant documents for prompt computation for $e_k \in D_i$ in the next steps.

Soft Prompt Computation: For convenience, let $\bar{e}_k$ be the representation for the event trigger e_k after the marker-augmented document $\bar{D}_i$ for D_i is encoded by the BART encoder in the previous step. As such, our soft prompt for EAE for trigger $e_k \in D_i$ will be a matrix P_{soft} of size $M_s \times d$ where M_s is a hyper-parameter and d is the dimension of the hidden vectors in our core model BART, thus allowing P_{soft} to be integrated into the computation of BART. To achieve a customized soft prompt P_{soft} for e_k in our model, the contextual representation $\bar{e}_k$ for e_k will be utilized to compute

P_{soft}. In addition, as discussed above, P_{soft} will also be conditioned on the aggregation of relevant context representations $\overline{et}$ for the the event type et of e_k to enrich the prompt and facilitate argument extraction. To this end, we utilize a learnable feed-forward network FF to transform the concatenation $\bar{e}_k$ and $\bar{t}_k^L$ into a vector of size $M_s \times d$. This vector will then be reshaped to form our soft prompt $P_{soft} = \text{reshape}(FF([\bar{e}_k; \overline{et}]))$.

Prompt-based EAE: While soft prompts enable example customization and relevant context incorporation for the models, we further inherit the hard prompts to explicitly specify expected argument roles for span extraction. In particular, to achieve a fair comparison, we utilize the same hard prompt for each event type as in previous work (Li et al., 2021; Ma et al., 2022) that connects all argument roles with natural language. For example, for the event type *Life.Consume.Unspecified*, the hard prompt to indicate argument roles is: *<ConsumingEntity> consumed <ConsumedThing> at <Place> place*.

For each event type t, we send its hard prompt into the embedding layer of BART to obtain a representation for each word (i.e., using averages of embeddings for sub-tokens), leading to the hard prompt representation P^t_{hard} of size $M^t_h \times d$ (M^t_h is the length of the hard prompt for t). We then concatenate the soft and hard prompt representations to create a single prompt Pr for EAE with BART, i.e., $Pr^t = [P^t_{hard}, P_{soft}]$ of size $(M_s + M^t_h) \times d$. Next, we follow the prompt-for-extraction framework in (Ma et al., 2022) to use the BART encoder to encode input context for D_i while the prompt P^t will be employed to prompt the BART decoder for span extraction. Given the current trigger $e_k \in D_i$, we first inject the trigger markers $<ET_i>$ and $</ET_i>$ before and after e_k in D_i to create an input context D'_i, which is then encoded by the BART encoder to return a sequence of representations D^{enc}_i for the words in D'_i. In the next step, the BART decoder is employed to learn richer representations for the context and prompt using their interactions via cross-attention in multiple layers, returning the representations $D^{dec}_i = \text{BART-Decoder}(D^{enc}_i; D^{enc}_i)$ and $\overline{Pr}^t = \text{BART-Decoder}(Pr; D^{enc}_i)$ for the context and prompt.

Afterward, for the j-th argument role for event type t, we obtain a role representation ϕ^t_j by mean-pooling its corresponding sub-token representations in the prompt representation $\overline{Pr}^t$. Similar to (Ma et al., 2022), we employ two selection heads s^{start} and s^{end} (of d dimensions) to compute start and end span selectors $\phi^{t,start}_j = \phi^t_j \odot s^{start}$ and $\phi^{t,end}_j = \phi^t_j \odot s^{end}$ ($\odot$ is the element-wise multiplication). Each span selector tuple $\theta^t_j = (\phi^{t,start}_j, \phi^{t,end}_j)$ then aims to select at most one argument span for the j-th role of t. Here, the golden span for this role is denoted by $(a^{t,start}_j, a^{t,end}_j)$. It will be set to $(0, 0)$ if event e_k does not have an argument of this role in D_i. As such, the extractive prompt framework is utilized to estimate distributions over token positions in D for how likely each token in D_i would serve as the start/end position for the argument span of the role: $P^{t,start}_j = \text{softmax}(\phi^{t,start}_j D^{dec}_i)$, $P^{t,end}_j = \text{softmax}(\phi^{t,end}_j D^{dec}_i)$. Finally, to train our model, we optimize the loss: $\mathcal{L} = -\sum_{e_k \in \mathcal{D}} \sum_j (\log(P^{t,start}_j(a^{t,start}_j)) + \log(P^{t,end}_j(a^{t,end}_j)))$ (i.e., over all events in $\mathcal{D}$).

Inference: At inference time, given an input text, event type, and argument role, we consider all possible argument spans for the role, ensuring that the start indexes are smaller than the end indexes (including $(0, 0)$) and their lengths do not exceed a maximum value computed over training data. A score for each span is obtained using the probability $\log(P^{t,start}_j(a^{t,start}_j)) + \log(P^{t,end}_j(a^{t,end}_j))$. The span with the highest score will be chosen for prediction. Finally, the aggregations $\overline{et}$ of relevant context representations for event types, which are learned during training, are used in test time.

3 Experiments

Datasets and Hyper-parameters: Following previous work (Li et al., 2021; Ma et al., 2022), we employ two latest datasets for EAE to evaluate our model, i.e., RAMS (Ebner et al., 2020) and WIKIEVENTS (Li et al., 2021). Both datasets involve multiple events in a document where arguments can distribute over different sentences from the event triggers. We utilize the same train/dev/test splits, data pre-processing, and evaluation metrics for the datasets as in previous work (Ma et al., 2022) for fair comparison. In particular, our metrics include: Argument Identification F1 score (Arg-I) and Argument Classification F1 score (Arg-C) scores. For WIKIEVENTS, we also use Argument Head F1 score (Head-C) to only consider headword matching for arguments. Finally, we fine-tune the hyper-parameters for our model on the development data.

Comparison: We compare our method (called **SPEAE** for soft prompts for EAE) with the state-of-the-art models for EAE. In particular, we consider two groups of baselines: (i) text generation-based models: BART-Gen (Li et al., 2021), and (ii) question answering-based models: FEAE (Wei et al., 2021), DocMRC (Liu et al., 2021a), EEQA (Du and Cardie, 2020), EEQA-BART (Du and Cardie, 2020), EA2E (Zeng et al., 2022), and PAIE (Ma et al., 2022). The performance of EA2E is obtained by running the provided code over our preprocessed data using the same evaluation metrics for a fair comparison. The performance for other baselines is inherited from (Ma et al., 2022), which presents the model PAIE with current best-reported results for our datasets.

Model	PLM	RAMS		WIKIEVENTS		
		Arg-I	Arg-C	Arg-I	Arg-C	Head-C
BART-Gen	BART-b	50.9	44.9	47.5	41.7	44.2
	BART-l	51.2	47.1	66.8	62.4	65.4
FEAE	BERT-b	53.5	47.4	-	-	-
DocMRC	BERT-b	-	45.7	-	43.3	-
EEQA	BERT-b	46.4	44	54.3	53.2	56.9
	BERT-l	48.7	46.7	56.9	54.5	59.3
EEQA-BART	BART-b	49.4	46.3	60.3	57.1	61.4
	BART-l	51.7	48.7	61.6	57.4	61.3
EA2E	BART-b	-	-	64.5	58.6	61.7
	BART-l	-	-	70.8	65.7	67.8
PAIE	BART-b	54.7	49.5	68.9	63.4	66.5
	BART-l	56.8	52.2	70.5	65.3	68.4
SPEAE (ours)	BART-b	**56.0**	**51.1**	**70.6**	**66.2**	**69.6**
	BART-l	**58.0**	**53.3**	**71.9**	**66.1**	**68.8**

Table 1: Model performance on test data.

Table 1 shows the performance of the methods over the test datasets along with their corresponding PLM versions. The most important observation from the table is that the proposed method SPEAE significantly outperforms the baseline methods (with $p < 0.01$) for both base and larger versions of the PLM models (i.e., BERT and BART). It just clearly demonstrates the benefits of the proposed method for EAE with contextualized soft prompts for instances and relevant context.

Model	**Arg-I**	**Arg-C**	**Head-C**
SPEAE (full)	70.6	66.2	69.6
No example context $\overline{e}_k$ for P_{soft}	69.7	65.5	68.4
No relevant context $\overline{et}$ for P_{soft}	70.1	65.8	68.9
No soft prompt P_{soft}	69.4	64.7	67.5
No graph for $\overline{et}$	69.1	65.3	68.1

Table 2: Ablation study on test data.

Ablation Study: To reveal the contribution of the designed components in SPEAE, we perform an ablation study over the WIKIEVENT test data. Table 2 presents the performance of the ablated models. In particular, for soft prompts, we first exclude either the example-specific context representation $\overline{e}_k$ or the relevant context aggregation $\overline{et}$ from the computation of the soft prompt P_{soft}. As the performance of the resulting models is significantly worse than SPEAE, it clearly testifies to the importance of such components for prompt-based models for EAE. The performance is furthered degraded when the soft prompt P_{soft} is completely eliminated from the prompt, thus suggesting the effectiveness of soft prompts for EAE. Additional, instead of computing the relevant context aggregations for event types $\overline{et}$ with a graph neural network, we explore a variant to directly obtain $\overline{et}$ from the average representation of the documents in $\mathcal{D}$ that contain an event mention of type et. The worse performance of the ablated model clearly confirms the benefits of the graph neural network for representation aggregation of relevant documents/event types for soft prompt computation for EAE.

Low-resource Learning: To better understand the operation of the proposed model SPEAE under low-resource training settings, we perform an evaluation when different ratios of training data are employed to train the models. In particular, we compare SPEAE with the previous state-of-the-art models, i.e., EEQA (Du and Cardie, 2020), EEQA-BART (Du and Cardie, 2020), and PAIE (Ma et al., 2022) in this low-resource learning experiment. Table 3 demonstrates the performance of the models (based on Arg-C) on the development data of WIKIEVENTS. As can be seen from the table, the proposed model SPEAE is significantly better than the baseline methods over different ratios of training data, ranging from 1% to 50%. It just clearly highlights the advantages of our proposed method for low-resource learning settings. We attribute these advantages to the introduction of context information from current example and relevant documents to enrich soft prompts, allowing SPEAE to better utilize available training data to boost performance.

Stability Study: One of the major issues with the discrete prompts in previous EAE models is that model performance can be sensitive to specific formats of the hand-designed prompts (Zhao et al., 2021; Ma et al., 2022). This raises an important concern for the applications of EAE models to different datasets and problems as optimal formats of the prompts might be unclear for new datasets,

Model	Training Data Ratio					
	1%	2%	5%	10%	20%	50%
EEQA	15.0	18.1	35.7	43.2	45.5	49.6
EEQA-BART	21.2	18.3	42.9	44.3	54.1	56.8
PAIE	31.3	40	52.1	51.4	54.9	59.8
SPEAE (ours)	**35.0**	**43.8**	**52.3**	**56.7**	**58.7**	**64.7**

Table 3: Low-resource learning performance (Arg-C) of the models on the development data of WIKIEVENTS. Models are trained on different ratios of training data.

necessitating further laborious efforts for prompt development and selection. To understand the sensitivity/stability of EAE models over different formats of discrete prompts, this experiment explores three variants of discrete prompts for EAE as discussed in (Ma et al., 2022), i.e., Manual Template, Uncontextualized Soft Prompt, and Concatenate Template. In particular, Manual Template (MA) involves the discrete prompts we utilize in our work, (Li et al., 2021), and (Ma et al., 2022). It concatenates all argument roles for an event type using natural language. For Uncontextualized Soft Prompt (USP), the prompts link argument roles with role-specific special tokens (Qin and Eisner, 2021; Liu et al., 2021b). These tokens are associated with learnable embedding vectors to help transform discrete prompts into representation vectors for further computation. Here, a key difference between these embeddings for argument role-specific tokens and our soft prompts is that our soft prompts are contextualized over current example context and relevant documents. In contrast, the learnable embeddings for role-specific tokens in USP are only initialized randomly, thus unable to contextualize over example context for better customization and richer prompts as in our soft prompts. Finally, in Concatenate Template (CA), all argument role names for an event type are simply concatenated to form prompts (Ma et al., 2022).

Using three variants of discrete prompts, we compare the performance (based Arg-C) of our proposed model SPEAE and the current state-of-the-art discrete-prompt model PAIE for EAE. Table 4 presents model performance on the RAMS development data. It is clear from the table that the proposed model SPEAE performs significantly better than PAIE over different variants of discrete prompts, thus further demonstrating the benefits of SPEAE. Importantly, while PAIE exhibits diverse performance gaps across different discrete prompts, SPEAE maintains more stable performance. This suggests an important advantage of SPEAE that is less sensitive to specific discrete prompt formats to enable convenient extension to new applications with less development efforts for prompt design.

Model	MA	USP	CA
PAIE	48.8	47.4	45.2
SPEAE (ours)	**51.5**	**52.2**	**51.1**

Table 4: Model performance over the development data of RAMS using different variants of discrete prompts: MA (Manual Template), USP (Uncontextualized Soft Prompts), and CA (Concatenate Template). Performance for PAIE is obtained by running the provided code from the original paper to achieve fair comparison.

4 Related Work

Multiple methods have been introduced to solve EAE, including early feature-based methods (Li et al., 2013; Yang and Mitchell, 2016) and recent deep learning models (Chen et al., 2015; Nguyen et al., 2016; Liu et al., 2018; Lin et al., 2020; Nguyen et al., 2022). While most previous methods have focused on the classification frameworks (Pouran Ben Veyseh et al., 2020; Nguyen et al., 2021; Pouran Ben Veyseh et al., 2022b), PLMs has enabled recent formulation of EAE via question answering (Du and Cardie, 2020; Liu et al., 2020; Li et al., 2020; Liu et al., 2021a; Wei et al., 2021) or text generation (Paolini et al., 2021; Li et al., 2021; Lu et al., 2021; Zeng et al., 2022) paradigms. At the core of such models involves questions/prompts to specify argument roles to prompt PLMs. However, the questions/prompts in previous methods are mainly discrete and hand-designed, making it hard to customize for each example and incorporate various relevant context.

5 Conclusion

We introduce a new prompt-based method for EAE that features soft prompts to achieve example customization and relevant context augmentation to enrich prompts. Extensive experiments demonstrate the advantages of the proposed method for EAE. In the future, we will explore soft prompts for other problems in IE for better understanding.

Limitations

In this work, we propose a novel method for EAE that introduces learnable soft prompts to capture specific-example context and relevant documents for prompt customization and enrichment. Although experiment results have demonstrated the

benefits of the proposed model, there are several limitations that can be addressed for further improvement in future work. First, similar to previous EAE studies (Du and Cardie, 2020; Li et al., 2021; Ma et al., 2022), our EAE model assumes golden event triggers for event types that might not be available for real-world applications. As such, future work can develop more comprehensive research and models to accommodate predicted event triggers while still maintaining competitive performance for EAE. Second, to aggregate relevant document representations for soft prompt computation, our EAE method leverage an event type mentioning graph that capture documents, event types, and their occurrence in training data. On the one hand, the graph does not involve argument roles that are directly related to EAE and might provide richer information/context to obtain representation aggregation to augment soft prompts. On the other hand, our method only explores graph attention networks to perform representation aggregation while many other variants of graph neural networks have not been considered, e.g., deep graph convolutional networks (Chen et al., 2020). Future work can explore richer graphs and graph neural networks to learn better representations for soft prompts for EAE. Third, despite the introduction of soft prompts with important benefits, our method still needs to rely on discrete prompts to explicitly specify event types and argument roles. Although our experiments demonstrate better stability of the proposed method with different discrete prompt variants, adapting our method to new languages will still require some prompt development effort to achieve optimal performance. Finally, in contrast to the interpretability of discrete prompts, soft prompts are less explainable, which can be addressed in future work to make the proposed method more accessible to various users.

Acknowledgement

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112, the NSF grant CNS-1747798 to the IUCRC Center for Big Learning, and the NSF grant # 2239570. This research is also supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract 2022-22072200003. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. 2020. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, pages 1725–1735. PMLR.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

Xinya Du and Heng Ji. 2022. Retrieval-augmented generative question answering for event argument extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xinya Du, Sha Li, and Heng Ji. 2022. Dynamic global memory for document-level argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5264–5275, Dublin, Ireland. Association for Computational Linguistics.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020. Event

extraction as multi-turn question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 829–838, Online. Association for Computational Linguistics.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online. Association for Computational Linguistics.

Jian Liu, Yufeng Chen, and Jinan Xu. 2021a. Machine reading comprehension as data augmentation: A case study on implicit event argument extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2716–2725, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *ArXiv*, abs/2103.10385.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. 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 2795–2806, Online. Association for Computational Linguistics.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6759–6774, Dublin, Ireland. Association for Computational Linguistics.

Minh Van Nguyen, Viet Dac Lai, and Thien Huu Nguyen. 2021. Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 27–38, Online. Association for Computational Linguistics.

Minh Van Nguyen, Bonan Min, Franck Dernoncourt, and Thien Nguyen. 2022. Learning cross-task dependencies for joint extraction of entities, events, event arguments, and relations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9349–9360, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics.

Trung Minh Nguyen and Thien Huu Nguyen. 2018. One for all: Neural joint modeling of entities and events. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*.

Amir Pouran Ben Veyseh, Javid Ebrahimi, Franck Dernoncourt, and Thien Nguyen. 2022a. MEE: A novel multilingual event extraction dataset. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9603–9613, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Minh Van Nguyen, Franck Dernoncourt, Bonan Min, and Thien Nguyen. 2022b. Document-level event argument extraction via optimal transport. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1648–1658, Dublin, Ireland. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Graph transformer networks with syntactic and semantic structures for event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3651–3661, Online. Association for Computational Linguistics.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks. *6th International Conference on Learning Representations*.

Kaiwen Wei, Xian Sun, Zequn Zhang, Jingyuan Zhang, Guo Zhi, and Li Jin. 2021. Trigger is not sufficient: Exploiting frame-aware knowledge for implicit event argument extraction. 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 4672–4682, Online. Association for Computational Linguistics.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299, San Diego, California. Association for Computational Linguistics.

Qi Zeng, Qiusi Zhan, and Heng Ji. 2022. EA^{2}E: Improving consistency with event awareness for document-level argument extraction. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2649–2655, Seattle, United States. Association for Computational Linguistics.

Tony Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the International Conference on Machine Learning (ICML)*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Please see Section 6 in the paper.

☒ A2. Did you discuss any potential risks of your work?
We do not observe significant risks in our work.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Please see the abstract and Section 1 for introduction of the paper.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Please see Section 3, Appendix A, and Appendix C.

☑ B1. Did you cite the creators of artifacts you used?
Please see Section 3 and Appendix A.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The datasets in our paper are publicly accessible. Appendix C discusses license of the libraries we used for model implementation.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Please see Section 3, Appendix A, and Appendix C.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Our datasets are public and widely used in previous research on Event Argument Extraction. There have been no concerns for private information or offensive content in our datasets in multiple previous work.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Please see Section 3 and Appendix A.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Please see Section 3 and Appendix A.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Please see Appendix C.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Please see Appendix C.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Please see Appendix C.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Please see Appendix C.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Please see Section 3 and Appendix C.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.