

## Article

# Role Knowledge Prompting for Document-Level Event Argument Extraction

Ruijuan Hu , Haiyan Liu and Huijuan Zhou

PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China

\* Correspondence: huruijuan01@126.com

**Abstract:** Document-level event argument extraction (DEAE) aims to identify the arguments corresponding to the roles of a given event type in a document. However, arguments scattering and arguments and roles overlapping make DEAE face great challenges. In this paper, we propose a novel DEAE model called Role Knowledge Prompting for Document-Level Event Argument Extraction (RKDE), which enhances the interaction between templates and roles through a role knowledge guidance mechanism to precisely prompt pretrained language models (PLMs) for argument extraction. Specifically, it not only facilitates PLMs to understand deep semantics but also generates all the arguments simultaneously. The experimental results show that our model achieved decent performance on two public DEAE datasets, with 3.2% and 1.4% F1 improvement on Arg-C, and to some extent, it addressed the overlapping arguments and roles.

**Keywords:** event argument extraction; document-level; prompting; role knowledge



**Citation:** Hu, R.; Liu, H.; Zhou H. Role Knowledge Prompting for Document-Level Event Argument Extraction. *Appl. Sci.* **2023**, *13*, 3041. <https://doi.org/10.3390/app13053041>

Academic Editor: Monica Gallo

Received: 26 December 2022

Revised: 21 February 2023

Accepted: 25 February 2023

Published: 27 February 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Event extraction is an extremely challenging information extraction task [1–3] that extracts events and corresponding arguments from a text. It provides effective structured information for knowledge graph construction [4], recommendation systems [5,6], etc. Event extraction is usually divided into two subtasks: event detection, which identifies an event's trigger words, and event argument extraction (EAE), which identifies the event arguments corresponding to a given trigger word and their roles. As in Figure 1a, given the trigger word “annihilate” with the event type “conflict.attack.n/a.”, the EAE model can extract roles such as “attacker”, “target”, and “place” from sentence S1, and the arguments are “Russian air forces”, “ISIS oil tankers”, and “Aleppo province”. In Figure 1b, the trigger word is “airstrike”, and the arguments “target”, “attacker”, and “instrument” are scattered in sentences S2, S3, and S4 (called argument scattering). In particular, the same argument may play different roles in the event (called argument overlapping). For example, “Yemen” is both the target and place. Meanwhile, there are also cases where the same role corresponds to multiple arguments (called role overlapping). For example, there are other places aside from “Yemen”. Arguments scattering and arguments and roles overlapping make document-level EAE (DEAE) more challenging than sentence-level EAE (SEAE) [7].

The main EAE approach is to learn the pairwise information between the trigger words and arguments. Some scholars use this as a classification task to directly locate and extract trigger words and arguments in the text and then classify them. Many of these works solve this problem by using sequence labeling techniques based on neural networks [8–11]. However, such methods cannot effectively solve the problem of arguments and roles overlapping. Other studies converted this problem into machine reading comprehension or question-and-answer tasks [12–14], where a set question-and-answer template allows the model to find text spans of trigger words and arguments. However, such methods split the argument extraction into multiple tasks, where only one argument can be extracted at a time, and eventually, multiple rounds of the question-and-answer

task are required to be completed, with the results of the previous round being passed to the next round, a process that lacks interaction between arguments. The generative pretraining approach enables the model to generate all arguments simultaneously through the learning capability of the pretrained models (PLMs) [15,16] while incorporating prompt learning to stimulate the potential of the PLMs better. However, the lack of interaction between the prompt template and the roles leads to the templates not providing accurate prompting well.

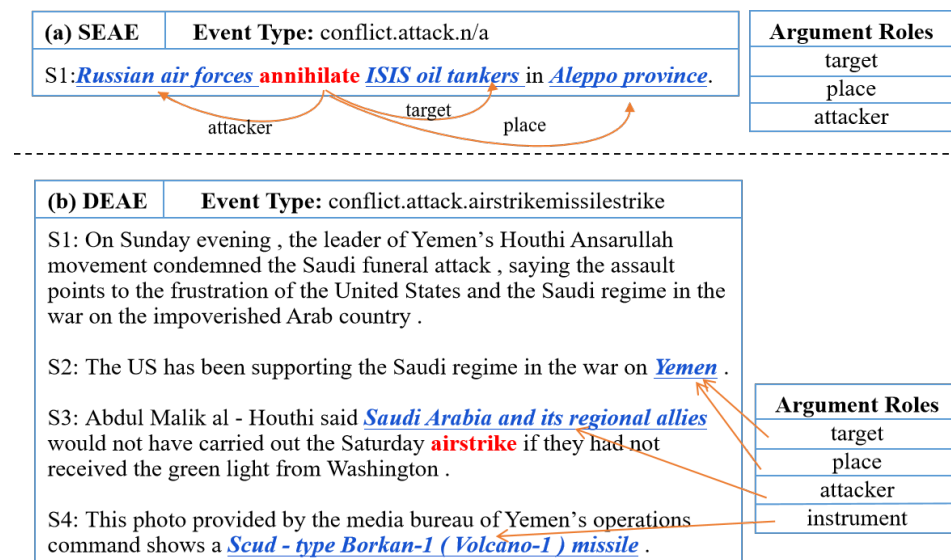


**Figure 1.** Illustrations of SEAE and DEAE. (a) An example of SEAE, where event arguments are extracted from a sentence. (b) An example of DEAE, where there are arguments scattering and overlapping. The trigger words are marked in red bold text, and the arguments are marked in blue italicized text with underlining.

Inspired by the fact that humans need prior domain knowledge to understand text, it is observed that role knowledge plays a pivotal role in understanding contextual semantics. As in the example in Figure 1b, in sentence S, “Scud-type Borkan-1 (Volcano-1) missile” is a certain type of missile, and if the meaning of the role “instrument” is known to be a tool, which refers to the tool used in an air strike event, then it is predicted simply that “Scud-type Borkan-1 (Volcano-1) missile” is an argument for the instrument in an air strike event. Similarly, as a city, “Yemen” is very likely to perform the role of the target or place. Therefore, it can be considered that role knowledge is the bridge between textual semantics and arguments.

Can we utilize role knowledge to guide prompting for sufficiently precise elicitation of PLMs and simultaneously extract all the arguments in events? On this basis, we propose a novel approach called role knowledge prompting for document-level event argument extraction (RKDE), which leverages role knowledge to guide templates for the precise prompting of PLMs. First, the DEAE task is modeled as a slot padding task that extracts all the arguments simultaneously, setting one slot for each argument to solve the argument scattering problem. Then, the prompt-tuning paradigm is introduced to identify the arguments by stimulating the ability of PLMs to address the arguments and roles overlapping problem. On the one hand, two sub-prompts, role definition and event description, are designed, where role definition aims to prompt the model in which information of the text should be focused and event description aims to provide a narrative description of the role in the event to help the model understand the semantics better. On the other hand, role knowledge is used to guide the template for prompting. Role concept knowledge and role interaction knowledge are incorporated into the template. They can help the model cope with arguments and roles overlapping by guiding the mechanism for template and role interaction. Finally, all the arguments are generated simultaneously by the text span

querier with a BART encoder-decoder model. The experimental results show that our model outperforms previous methods.

Our major contributions are summarized as follows:

- (1) We propose a role knowledge-prompting model for DEAE capable of extracting all arguments by slot span filling, activating PLMs' potential with a prompt-tuning paradigm to guide the template precisely.
- (2) We designed a role knowledge-prompting model to enhance the representation of roles through the interaction of the text, template, and role knowledge, which helps to better capture the deep semantic relationships. This gives our model a good ability to cope with arguments and roles overlapping.
- (3) The experimental results show that our method achieves the state of the art on two document-level benchmark datasets—RAMS and WIKIEVENTS—with a 3.2% and 1.4% F1 improvement in argument classification, respectively.

## 2. Related Work

**Document-level event argument extraction.** DEAE is an extremely challenging subtask of event extraction. DEAE can be traced back to the role-filling task from MUC conferences [17], which requires retrieval of the scenario-specific participating entities and attribute values. The TAC Knowledge Base Population slot filling challenge (<https://tac.nist.gov/2017/KBP/index.html>, accessed on 13 November 2017) is similar to this task. Traditional event extraction methods [18–22] rely on hand-crafted features and patterns. With the ongoing development of neural networks, further extracted event arguments were based on the distributional features [8,9,23–25]. Recently, zero-shot learning [26], multi-modal integration [27], and weakly supervised [28] methods have been adopted to improve event extraction.

EAE methods can be grouped into three categories. The first is the approach that perceives the context globally from the documents and classifies it by obtaining semantic representations at different granularities (word granularity, sentence granularity, and document granularity) [23,29–31]. Liu et al. used the local features of the arguments to assist in role classification [18]. Chen et al. proposed a dynamic multi-pooling convolutional neural network model [8]. With the development of deep learning, event extraction is a sequence labeling task [31–34], but these methods pale in comparison to an arbitrary location and deep semantic relations and cannot solve argument overlapping well. The second is to capture the unique features of a document using graph convolutional networks (GCNs) by modeling graph models such as syntactic semantic graphs, event connectivity graphs, and heterogeneous graphs to capture the interaction of arguments. The DBRNN enhanced the representation by bridging associated words on syntactic dependency trees [24]. Xu et al. constructed heterogeneous graphs of sentences and entity mentions to strengthen the interaction between sentences and entity mentions [35]. The third is to transform event extraction into tasks such as reading comprehension and multi-round question-and-answer tasks [36,37], formulating the roles as questions described in natural language and extracting the arguments by answering these questions in context. This makes better use of the a priori information of the roles' categories, but such methods require setting multiple relevant questions, and the model can only answer one question at a time.

As the generative pretrained models become more powerful, such as T5 [38] and BART [39], they show extraordinary text generation capabilities, and recently, several extraction jobs have been converted to be completed as generative tasks [40]. Li et al. tackled DEAE with an end-to-end neural network model for conditional text generation [16]. Text2Event put forth a sequence-to-structure generation paradigm that could extract events directly from text in an end-to-end manner [15]. In contrast to these approaches, our approach emphasizes the interaction between the arguments and can generate all the arguments simultaneously.

**Prompt tuning.** Different from the “pretraining + fine-tuning” paradigm, prompt learning involves adapting downstream tasks to PLMs, which can be regarded as a retrieval method for knowledge already stored in PLMs and has been widely used for text

classification [41], relation extraction [42], event extraction [43,44], and entity classification [45]. Researchers have made efforts to determine how to design prompt templates. Automatic searching of discrete prompts [46], gradient-guided searches [47], and continuous prompts including P-tuning [48] and prefix-tuning [49] were successively proposed. Recently, some studies have tried to integrate external knowledge into the prompt design. Tsinghua University proposed PTR [42], which implants logic rules into prompt tuning, and KPT [50], which extends label mapping through external knowledge graphs, achieving large performance gains in task scenarios. However, previous works [51,52] have shown that not all external knowledge can bring gains, and unselective implantation of external knowledge can sometimes introduce noise. Unlike these approaches, we add role-specific knowledge to guide the template for more precise prompting, compensating for the lack of role knowledge in the PLMs for EAE.

### 3. Methodology

RKDE considers role knowledge, joining templates to prompt PLMs for the simultaneous extraction of document-level event arguments. The model framework is shown in Figure 2. It consists of four stages: template design, role knowledge guidance, text span query, and argument generation. In this section, we will first introduce the DEAE problem definition and then describe each component of the model in turn.

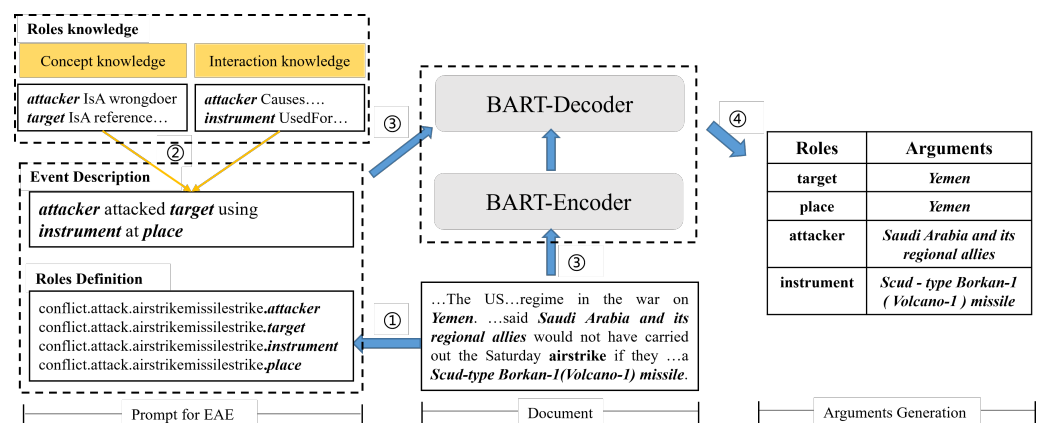


**Figure 2.** The overall framework of our RKDE model. Given a document, RKDE first creates an EAE prompt template based on known event types (① includes two parts: role definition and event description). This process uses role knowledge guidance (②) to facilitate more precise prompting of the template by using role concept knowledge and interaction knowledge. Then, the document and the prompt template will be fed into the BART encoder and the BART decoder, respectively (③), and the text span querier is used to find the argument span. Finally, the BART decoder is exploited for prediction (④) to generate the arguments corresponding to the roles.

#### 3.1. Problem Definition

For modeling event argument extraction as a text generation task, given a document  $D$ , the event type is  $e$ , the event trigger word is  $t \subseteq D$ , and the argument role  $r \in Role^{(e)}$ . For example, in the document in Figure 1, the event type  $e$  is “*conflict.attack.airstrikemissilestrike*”, and the event trigger word  $t$  is “*airstrike*”, where the argument role is denoted as

$$Role^{(e)} = \{attacker, target, instrument, place\} \quad (1)$$

Our task is to extract a series of spans  $A$  for a given  $(X, t, e, Role^{(e)}) \in D$  such that each role corresponds to an argument  $arg^r \in A$ .

#### 3.2. Template Design for DEAE

The purpose of the prompt template is to further motivate the PLMs to understand the contextual semantics. For the DEAE task, the prompt template’s goal is to make the model

understand the description of events and the argument roles to motivate it to generate the correct arguments. Therefore, the designed prompt template contains two parts, namely role definition and event description, which are formalized as

$$T_{EAE} = [T_{roledef}; T_{eventdes}] \quad (2)$$

where  $T_{roledef}$  denotes role definition,  $T_{eventdes}$  denotes event description, and  $T_{EAE}$  is the splicing of the two parts.

**Role definition.** The template should contain all roles for the event, so the role definition for the corresponding events that are already defined in the dataset is part of the template. This suggests which role information the model should be focused on. As shown in Figure 2, the role definition defines four roles associated with the event conflict.attack.airstrikemissilestrike:

$$\begin{aligned} T_{roledef} : & \text{conflict.attack.airstrikemissilestrike.attacker;} \\ & \text{conflict.attack.airstrikemissilestrike.target;} \\ & \text{conflict.attack.airstrikemissilestrike.instrument;} \\ & \text{conflict.attack.airstrikemissilestrike.place} \end{aligned} \quad (3)$$

**Event description.** Inspired by narrative writing and comprehension, we find that event descriptions such as “who did what, when, and where” are helpful for understanding the details of the event. Thus,  $Role^{(e)}$  was sorted into the event description text as the other part of the template. In Figure 2, the event description is designed as follows:

$$T_{eventdes} : \text{attacker attacked target using instrument at place} \quad (4)$$

The role position (underlined part) is set to a slot. To solve the role overlapping problem (i.e., the same role corresponds to more than one argument), multiple slots were set for the same role (i.e., a target matches with two slots):

$$T_{eventdes} : \text{attacker attacked target (target) using instrument at place} \quad (5)$$

### 3.3. Role Knowledge Guidance

The template specifies the concrete roles corresponding to the given events, along with the narrative descriptions. The model can initially determine which roles should be extracted from the text according to the prompts, but it does not understand the deeper meanings or the interactions between the roles. Therefore, to maximize the role of the template as a deeper and more accurate prompt, we designed a role knowledge guidance mechanism.

For role knowledge acquisition, a knowledge graph with a large variety of common knowledge, entity knowledge, and semantic relations is undoubtedly the best choice for external knowledge. ConceptNet [53] is a knowledge graph with rich concepts and semantic relations, containing 21 million edges, more than 8 million nodes, and 34 core relations. We retrieved the role knowledge contained in the event description  $T_{eventdes}$  in ConceptNet, which is mainly divided into two types: one is role concept knowledge, obtained from four semantic relations (*IsA*, *HasProperty*, *HasA*, and *PartOf*), and the other is role interaction knowledge or role relationship knowledge, obtained from six semantic relations (*CapableOf*, *Causes*, *CauseDesire*, *UsedFor*, *Desires*, and *HasPrerequisite*).

Specifically, we first matched the tokens of the roles in the nodes of ConceptNet and retrieved the semantic relations and associated nodes related to these tokens. For example, the concept knowledge of “attacker”, namely “attacker IsA wrongdoer”, was retrieved. Regarding the role guidance mechanism, external knowledge injection can introduce noise and thus affect the understanding of the semantics of the original text. To solve this problem, inspired by Ye et al. [54], we propose a role guidance mechanism that uses role knowledge

to guide the templates for more precise prompting on the premise of reducing the influence of irrelevant knowledge, as shown in Figure 3.

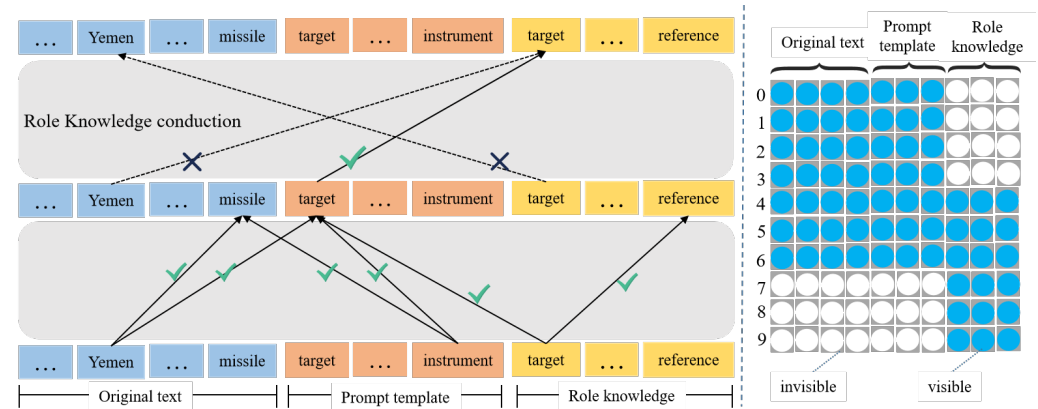


**Figure 3.** Role knowledge guidance mechanism. Interaction between original text, templates, and roles, where ✓ indicates visible and × indicates invisible.

To instruct the template for better comprehension of the roles, a role knowledge guidance mechanism was designed for the interaction between the original text, the template, and role knowledge. Specifically, the role knowledge is visible to the template, the text and the template are visible to each other, and the role is visible internally. For the purpose of interaction between the template and the role knowledge, the template is visible to the role, but the role and the original text are invisible to each other. The attention mask matrix is defined as

$$M_{ij} = \begin{cases} 0, & w_i, w_j \in x_D \oplus x_{T_{EAE}} \\ 0, & w_i, w_j \in x_{Role} \\ 0, & w_i \in x_{Role}, w_j \in x_{T_{EAE}} \\ 0, & w_i \in x_{T_{EAE}}, w_j \in x_{Role} \\ -\infty, & otherwise \end{cases} \quad (6)$$

where  $w_i$  and  $w_j$  denote any two tokens in the input sequence, 0 denotes that there is attention from tokens  $w_i$  to  $w_j$ ,  $-\infty$  denotes that there is no attention from  $w_i$  to  $w_j$ ,  $D$  denotes the original text,  $T_{EAE}$  denotes the template, and Role denotes the role knowledge. Role knowledge is a more complete and detailed description of the roles, which can contribute to the model's better understanding of the in-depth semantics of the text. It can influence and guide the template, and the potential relationships between roles can assist in solving the argument overlapping problem.

### 3.4. Argument Generation

Given the context  $X$  and the template  $T_{EAE}$ , for each role in  $T_{EAE}$  corresponding to a slot  $i$ , the model queries the text spans through a text span querier to generate the relevant event arguments.

#### 3.4.1. Text Span Querier

Given the input text  $X = (x_1, x_2, \dots, x_{trigger}, \dots, x_n)$ , we add the special token  $< t >$  and  $< /t >$  before and after  $x_{trigger}$ . The input text for the BART-Encoder is  $X' = (x_1, x_2, \dots, < t >, x_{trigger}, < /t >, \dots, x_n)$ . The template  $T_{EAE}$  is fed to the BART-Decoder. The hidden layer features after the encoder and decoder are represented as follows:

$$H(X) = \text{Encoder}(X') \quad (7)$$

$$H(T_{EAE}) = \text{Decoder}(H(X); T_{EAE}) \quad (8)$$

where  $H(X)$  denotes the contextual representation of a document after encoding and  $H(T_{EAE})$  denotes the representation of the EAE template of the document, with the text interacting with the template in the hidden layer of the decoder. Our goal was to obtain the range of the span corresponding to the slot of each role  $r$ . For any slot of role  $r_q$ , the role feature  $H(r_q)$  is obtained by  $H(T_{EAE})$  average pooling, and the range of the span was selected by the text span querier. The start and end of the span are indicated by  $start(r_q)$  and  $end(r_q)$ , respectively, and thus

$$start(r_q) = H(r_q) \cdot w^{start} \quad (9)$$

$$end(r_q) = H(r_q) \cdot w^{end} \quad (10)$$

where  $w^{start}$  and  $w^{end}$  are learnable parameters shared by all roles and “ $\cdot$ ” denotes element-wise multiplication, while  $textspan_i = [start(r_q), end(r_q)]$  is the  $q$ th slot querier in the template  $T_{EAE}$ . Regardless of whether the roles overlap, the text span querier can generate the argument for each slot, which can solve the previously mentioned role overlapping problem. At the same time, in this manner, the interaction between the text and the template through the cross-attention layer in the BART-Decoder enhances the prompting effect of the template on the text, improves the understanding of the deep semantics of the text, and realizes the information transfer between different roles.

### 3.4.2. Argument Generation

Based on the obtained contextual representation  $H(X)$  and the  $textspan_q$  for each role, the relevant argument span  $(s_q, e_q)$  is to be extracted from  $H(X)$ . If  $textspan_q$  is associated with an argument in  $H(X)$ , then  $arg_q = X'[i : j]$ , where  $i, j$  denotes the word subscript corresponding to the argument and the predicted output is  $(s_q, e_q) = (i, j)$ . If  $textspan_q$  is not associated with any argument in  $H(X)$ , then the predicted output is  $(s_q, e_q) = (0, 0)$ , which represents no argument corresponding to the role. Therefore, the set of candidate argument spans is

$$A = \{(i, j) | (i, j) \in L^2, 0 < (j - i) < m\} \cup \{0, 0\} \quad (11)$$

where  $L$  denotes the context length and  $m$  denotes the length threshold between the end and the beginning of the span (set to 10 in this paper). We calculated the probability distribution of each token being selected as the start or end of the argument span, using softmax as the activation function:

$$t_q^{start} = start(r_q)H(X) \quad (12)$$

$$t_q^{end} = end(r_q)H(X) \quad (13)$$

$$p_q^{start} = softmax(t_q^{start}) \quad (14)$$

$$p_q^{end} = softmax(t_q^{end}) \quad (15)$$

We define the loss function as

$$L_q(X) = \sum_{x \in D} \sum_q -(\log p_q^{start}(s_q) + \log p_q^{end}(e_q)) \quad (16)$$

In the inference stage, all candidate spans are enumerated and scored with the scoring function

$$Score_q(i, j) = t_q^{start}(i) + t_q^{end}(j) \quad (17)$$

Finally, according to the scoring function, the model predicts the span of the role  $r_q$  corresponding to the slot

$$(s_q, e_q) = \operatorname{argmax}_{(i,j) \in A} \operatorname{Score}_q(i, j) \quad (18)$$

## 4. Experiments

Our experiment aimed to verify (1) whether the prompting template-based generative model is effective for EAE and (2) whether role knowledge plays an effective role in guiding the template for more accurate prompting.

### 4.1. Datasets and Evaluation Metrics

**Datasets.** Our proposed method will be evaluated on two widely DEAE datasets: RAMS [7] and WIKIEVENTS [16]. RAMS is a document-level dataset annotated with 139 event types, 3194 documents, and 65 roles, and WIKIEVENTS is another document-level dataset whose documents were collected from English Wikipedia articles that describe real-world events, and then the corresponding links were followed to crawl related news articles. WIKIEVENTS provides 246 documents with 50 event types and 59 roles. We divided the datasets into a training set, validation set, and test set. The detailed division of the datasets is shown in Table 1.

**Table 1.** Datasets statistics. “Train” denotes training set, “Dev” denotes development set or validation set, and “Test” denotes test set.

| Datasets         | RAMS   | WIKIEVENTS |
|------------------|--------|------------|
| Sentences: Train | 7329   | 5262       |
| Sentences: Dev   | 924    | 378        |
| Sentences: Test  | 871    | 492        |
| Arguments: Train | 17,026 | 4552       |
| Arguments: Dev   | 2188   | 428        |
| Arguments: Test  | 2023   | 566        |

**Evaluation Metrics.** As in a previous work [16,55], we adopted two evaluation metrics: (1) the argument identification (Arg-I) F1 score, which evaluates whether the event arguments are detected or not by checking whether the generated arguments match the event type and roles, and (2) the argument classification (Arg-C) F1 score, which determines whether the event arguments are classified into the correct roles. For WIKIEVENTS, we added the head classification (Head-C) F1 score, which only cares about the first word matched to the argument but not all words.

### 4.2. Experimental Settings

In the implementations, we set all event types to be known, and only the event arguments were identified and classified. We chose the generative PLM BART-base as the encoder and decoder. We fed the documents to the BART-Encoder and the prompts to the BART-Decoder. Noting that the document length sometimes exceeded the encoder limit, a word window size centered on the trigger word was set as the input document selection. We trained the models with 5 fixed seeds (13, 21, 42, 88, and 100). The batch size was set to 4, with  $2 \times 10^{-5}$  as the learning rate with the Adam optimizer. The max span length was set to 10, and the window size was set to 250. The encoder’s maximum sequence length was set to 500, and the decoder’s maximum sequence length was set to 100.

### 4.3. Baselines

We compared our model with the state-of-the-art (SOTA) models for DEAE:

- SpanSel [7], a span ranking-based model, enumerates all possible spans in a document to identify event arguments.

- DocMRC [12] is an approach that models EAE as a machine reading comprehension (MRC) task, using BERT as a PLM.
- DocMRC(IDA) [12], which uses an implicit data augmentation approach for EAE, is an enhanced version of DocMRC.
- BERT-QA [13] is an approach that translates EAE into a QA task, using BERT as a PLM.
- BART-Gen [16] is an approach for generating the event arguments by sequence-to-sequence modeling and prompt learning, using the BART-large generative model as a PLM.

The implementation details of all baselines are as follows:

- SpanSel [7]: We report the results from the original paper.
- DocMRC [12]: We report the results from the original paper.
- DocMRC(IDA) [12]: We report the results from the original paper.
- BERT-QA [13]: We used their code (<https://github.com/xinyadu/eeqa>, accessed on 26 October 2020) to test its performance. We set the question template as “What is the **ROLE** in **TRIGGER WORD**?”, following the second template setting in the original paper.
- BART-Gen [16]: For the BART-base model, we used their code (<https://github.com/raspberrycice/gen-arg>, accessed on 13 April 2021) to test its performance. For the BART-large model, we report the results from the original paper.

#### 4.4. Main Results and Analysis

Table 2 shows the performance of our approach and all benchmark models on the RAMS and WIKIEVENTS datasets.

**Table 2.** Main results on RAMS and WIKIEVENTS datasets (F1 score (%)).

| Method                     | RAMS        |             | WIKIEVENTS  |             | Head-C      |
|----------------------------|-------------|-------------|-------------|-------------|-------------|
|                            | Arg-I       | Arg-C       | Arg-I       | Arg-C       |             |
| SpanSel [7]                | -           | 40.7        | -           | 38.3        | -           |
| DocMRC [12]                | -           | 44.3        | -           | 42.1        | -           |
| DocMRC(IDA) [12]           | -           | 45.7        | -           | 43.3        | -           |
| BERT-QA [13]               | 46.4        | 44.0        | 54.3        | 53.2        | 56.9        |
| BART-Gen (BART-Base) [16]  | 50.9        | 44.9        | 47.5        | 41.7        | 44.2        |
| BART-Gen (BART-Large) [16] | 51.2        | 47.1        | 66.8        | 62.4        | 65.4        |
| RKDE (ours)                | <b>55.1</b> | <b>50.3</b> | <b>69.1</b> | <b>63.8</b> | <b>66.8</b> |

**Main Results.** Overall, our proposed RKDE framework outperformed all the other methods on the test set. The performance of our proposed RKDE method was the best on both datasets, with the F1 score of Arg-C obtaining 3.2% and 1.4% improvement over the state-of-the-art model BART-Gen (BART-Large) on the RAMS and WIKIEVENTS datasets, respectively. We used BART-Base in the experiments. If compared with BART-Gen (BART-Base), the performance of our method on both data sets then exceeded it even more with the F1 score of Arg-C obtaining 5.4% and 22.1% improvement on the RAMS and WIKIEVENTS datasets, respectively. This shows the effectiveness of our method.

**Performance Analysis.** Concretely, we modeled the DEAE task as a slot padding task by extracting all arguments simultaneously, activating the PLM’s potential with a role knowledge prompt-tuning paradigm to guide the template precisely:

- (1) When comparing the approaches for the MRC and QA tasks (where MRC means machine reading comprehension while QA means question and answer), MRC is usually in the form of fill-in-the-blank completion for the model to determine which hidden word is most likely, depending on the context, while the questions of QA are answered one by one in accordance with the set question templates. For DEAE, these approaches both use the BERT model, which is applicable to natural language understanding. DocMRC and BERT-QA had comparable performance on the RAMS dataset, but

BERT-QA had a substantial improvement over DocMRC on the WIKIEVENTS dataset. Our approach is similar to that of reading comprehension, which is carried out in the form of span extraction and has a nearly 10 point performance improvement over the former. We conjecture that the performance improvement mainly lies in adding templates to the pretrained model.

- (2) In comparing the approaches of generative tasks, the generative model BART-Gen and our model both used prompt templates and had similar performances, but our performance was a bit better. We found that role knowledge guidance plays an active role in the templates. Unlike the prompt of BART-Gen, we designed two sub-prompts (i.e., role definition and event description) to help the model understand the semantics better and introduce role concept knowledge and role interaction knowledge into the template to guide the template for more precise prompting. The results demonstrate that role knowledge prompting for DEAE not only helps to locate the arguments accurately, but it also can cope with arguments and roles overlapping.

#### 4.5. Ablation Experiments

To analyze how each component in the proposed RKDE model contributed to the performance, we conducted ablation studies to turn them off one at a time, as Table 3 shows:

- (1) Without eventdes: Removing the event description from the template resulted in a 2.1% and 5.8% decrease in the F1 score of Arg-C on the two respective datasets. Intuitively, the description of the event arguments according to the logic of event occurrence enhanced the semantic expressiveness of the template.
- (2) Without rolekg: To verify the effectiveness of the role knowledge guidance module, we removed the role knowledge module and used only templates for prompting. Without the role concept knowledge to add descriptions to the roles in the template, and without the role interaction knowledge to guide the role relationships, the performance of the model decreased by 0.8% and 1.4% on Arg-C F1. This shows the importance of role knowledge, and it acts as a guide for the prompt.
- (3) Without prompt: To demonstrate the necessity of the prompt template module, the entire prompt was removed, and DEAE was performed with only the original text as input without giving any prompt to the model. This resulted in a significant decline in performance. This illustrates that the prompt templates can stimulate the learning ability of the PLMs and the precise prompting promotes more accurate prediction of the model.

**Table 3.** Ablation experiments, with main results on RAMS and WIKIEVENTS (F1 score (%)).

| Method           | RAMS  |       |                        | WIKIEVENTS |       |        |                        |
|------------------|-------|-------|------------------------|------------|-------|--------|------------------------|
|                  | Arg-I | Arg-C | $\nabla(\text{Arg-C})$ | Arg-I      | Arg-C | Head-C | $\nabla(\text{Arg-C})$ |
| RKDE (ours)      | 55.1  | 50.3  | -                      | 69.1       | 63.8  | 66.8   | -                      |
| Without eventdes | 54.6  | 48.2  | -2.1                   | 63.7       | 57.6  | 60.9   | -5.8                   |
| Without rolekg   | 54.7  | 49.5  | -0.8                   | 68.9       | 62.4  | 65.5   | -1.4                   |
| Without prompt   | 54.0  | 47.6  | -2.7                   | 63.2       | 57.3  | 60.1   | -6.5                   |

With the ablation experiments, it was observed that all the modules contributed to the model performance, and the role knowledge-guided templates could enhance the model's deep understanding of text semantics and accurately prompt the model for event argument generation, which led to a significant effect on DEAE.

#### 4.6. Impact of the Input Encoder Content

Our RKDE approach was to feed the document to the BART-Encoder and the prompt to the BART-Decoder. Can the document and the prompt be stitched together and fed directly to the encoder? We conducted experiments, as shown in Figure 4, and found that

the performance of stitching and feeding the prompt to the encoder had little degradation compared with the former. We considered that although the stitched text was ostensibly interacting with the template and the document, such prompting does not work well in longer documents to improve semantic understanding.

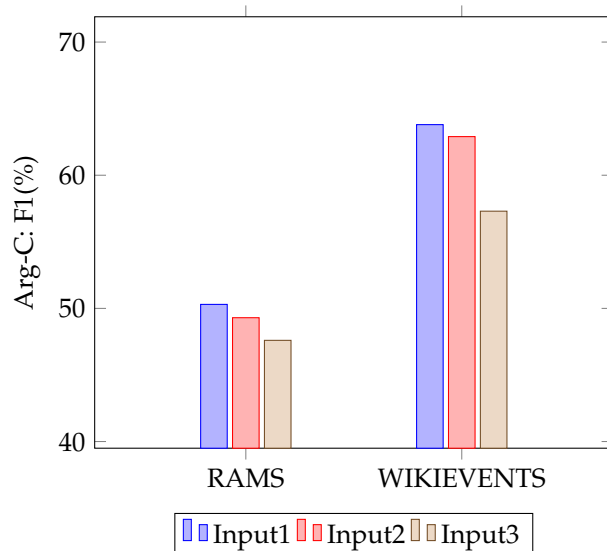


**Figure 4.** The impact of input encoder content, where Input1 indicates our method (RKDE), where we input the document into the encoder and the prompt into the decoder. Input2 indicates that the document and prompt were spliced into the encoder. Input3 indicates that the document was input into the encoder while not using the prompt.

#### 4.7. Effectiveness of Role Knowledge

To further verify the effectiveness of role knowledge for the model, we compared the performance in terms of accuracy at different steps with and without the injecting of role knowledge, as shown in Figure 5. It can be seen that the accuracy of the model with role knowledge (+role) was basically higher than that of the model without role knowledge (-role) at each step in both datasets, proving that role knowledge does indeed exert a certain influence on the DEAE task.

#### 4.8. Case Study

To visually demonstrate the effectiveness of each component of the role knowledge prompting approach, we performed a case study, as shown in Table 4.

Case 1 involved four arguments. All methods could correctly extract the argument giver, but for the roles of recipient and beneficiary, the method without a prompt could not identify it, while using the +role method correctly identified it. Noting that there was a problem of arguments overlapping, police was both the giver and beneficiary. The method without a prompt could not do anything about this phenomenon. Case 2 also illustrates that the method of using role knowledge prompting could enable the model to precisely locate the arguments corresponding to the roles based on understanding the deep semantics. Case 3 had the phenomenon of roles overlapping, where the role participant responded to three different arguments, and from the results, we found a strange observation: the method without a prompt could only identify one of the arguments, and the method with role knowledge identified two arguments without identifying the government. Meanwhile, the method with TEAE extracted all the arguments. We analyzed the reason for this and found that TEAE was designed as participant (and participant) communicated with participant (and participant) about topic (and topic) at place (and place), with a corresponding slot for each argument so that the model could extract the arguments with TEAE. However, the added role knowledge was the participant IsA associate. The template also required

comprehension of the associate, which confused the semantics in this case. Overall, role knowledge still contributed to performance improvement.

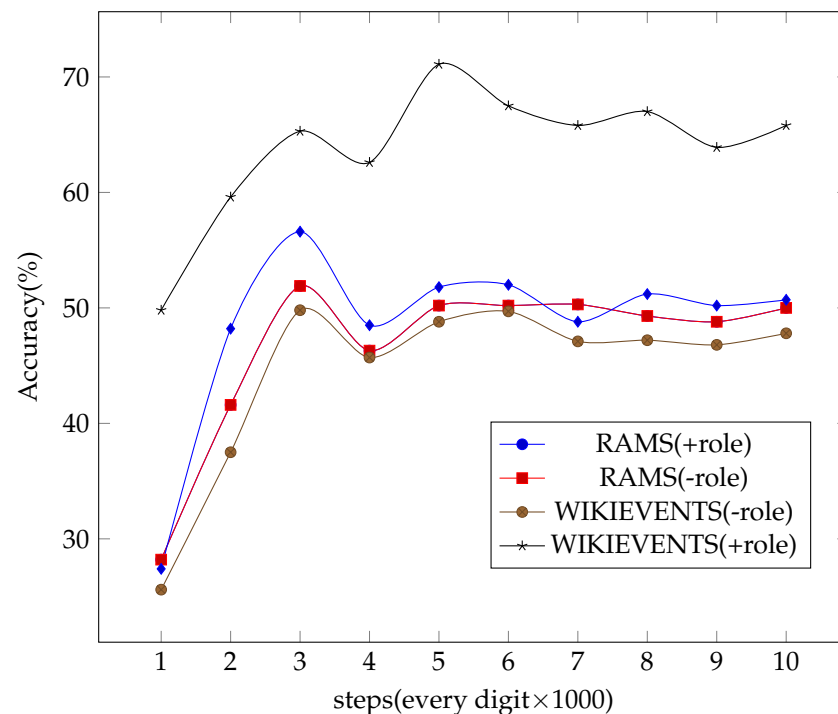


**Figure 5.** Effectiveness of role knowledge on model accuracy. RAMS(+role) and RAMS(-role) denote models with and without role knowledge on dataset RAMS, respectively, and WIKIEVENTS(+role) and WIKIEVENTS(-role) denote models with and without role knowledge on dataset WIKIEVENTS, respectively.

**Table 4.** Case study. Bold black is the trigger word, and bold black with underlining is the argument. “Nop” means the model did not have a prompt. “TEAE” means using a prompt template containing the role definition and event description, and “+role” means using a prompt with role knowledge guidance.

| No. | Samples                                                                                                                                                                                                                                                                                                                                               | Event Type                                       | Role: Argument                                        | Nop | TEAE | +Role |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|-------------------------------------------------------|-----|------|-------|
| 1   | According to investigators, police seized a sum of some \$120 million and €2 million (\$2.2 million). “The final amount is unknown. <u>Police</u> <t> <b>confiscated</b> </t> <u>a cache</u> only in <u>Zakharchenko’s house</u> ; it is not yet known how many more assets he has, but we are working in this direction,” ....                       | Transaction.<br>Transaction.<br>Transfercontrol. | <b>Recipient:</b> a cache                             | ✗   | ✓    | ✓     |
|     |                                                                                                                                                                                                                                                                                                                                                       |                                                  | <b>Beneficiary:</b> Police                            | ✗   | ✗    | ✓     |
|     |                                                                                                                                                                                                                                                                                                                                                       |                                                  | <b>Giver:</b> Police                                  | ✓   | ✓    | ✓     |
|     |                                                                                                                                                                                                                                                                                                                                                       |                                                  | <b>Place:</b> Zakharchenko’s house                    | ✗   | ✗    | ✗     |
| 2   | Merkel has faced growing criticism in her country for allowing an unprecedented number of asylum seekers (almost 1.1 million last year) to enter <u>Germany</u> . In her weekly video podcast Saturday, <u>she</u> <t> <b>urged</b> </t> <u>refugees from Iraq and Syria</u> to integrate in the country and learn German ...                         | Contact.<br>Requestadvise.<br>Correspondence.    | <b>Place:</b> Germany                                 | ✗   | ✓    | ✓     |
|     |                                                                                                                                                                                                                                                                                                                                                       |                                                  | <b>Recipient:</b> refugees from Iraq and Syria        | ✗   | ✗    | ✓     |
|     |                                                                                                                                                                                                                                                                                                                                                       |                                                  | <b>Communicator:</b> she                              | ✓   | ✓    | ✓     |
| 3   | In February 1975, the <u>Provisional Irish Republican Army</u> <t> <b>agreed</b> </t> to a truce and ceasefire with the British <u>government</u> and the <u>Northern Ireland Office</u> . Seven “incident centres” were established in Irish nationalist areas in Northern Ireland to monitor the ceasefire and the activity of the security forces. | Contact.<br>Contact.<br>Unspecified.             | <b>Participant:</b> Provisional Irish Republican Army | ✓   | ✓    | ✓     |
|     |                                                                                                                                                                                                                                                                                                                                                       |                                                  | <b>Participant:</b> government                        | ✗   | ✓    | ✗     |
|     |                                                                                                                                                                                                                                                                                                                                                       |                                                  | <b>Participant:</b> Northern Ireland Office           | ✗   | ✓    | ✓     |

## 5. Conclusions

In this paper, we proposed a novel approach: role knowledge prompting for document-level event argument extraction (RKDE). To make full use of the PLMs' abundant prior knowledge and learning capability, we designed prompt templates containing role definitions and event descriptions and introduced a role knowledge guidance mechanism to enhance the interaction between roles and templates for precise prompting of models. In the encoding and decoding phase, a text span querier was used to locate the spans, and eventually, all the arguments were generated simultaneously. The experimental results on two widely used DEAE datasets show that our approach somewhat solves the challenges of arguments and roles overlapping, and it had good performance in DEAE tasks. However, the prompt templates were still manually constructed and relied heavily on experience. In future work, we will consider how to automatically generate prompt templates or even replace manually constructed hard prompts with soft prompts containing continuous embedding for DEAE, and we will further deliberate about both the choice of role knowledge and how to guide the templates more precisely.

**Author Contributions:** Conceptualization, R.H.; Methodology, R.H.; Software, R.H.; Validation, R.H.; Formal analysis, R.H.; Investigation, H.L.; Resources, H.Z.; Data curation, H.Z.; Writing—original draft, H.Z.; Writing—review & editing, H.L.; Visualization, H.Z.; Supervision, H.L.; Project administration, H.L.; Funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Social Science Foundation of China (2022-SKJJ-B-066).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. RAMS data can be found here: [<https://nlp.jhu.edu/rams/>]. WIKIEVENTS data can be found here: [<https://github.com/raspberryice/gen-arg>]

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## Abbreviations

The following abbreviations are used in this manuscript:

|      |                                                                       |
|------|-----------------------------------------------------------------------|
| RKDE | Role knowledge prompting for document-level event argument extraction |
| PLMs | Pretrained language models                                            |
| EE   | Event extraction                                                      |
| EAE  | Event argument extraction                                             |
| SEAE | Sentence-level event argument extraction                              |
| DEAE | Document-level event argument extraction                              |

## References

1. Liu, K.; Chen, Y.; Liu, J.; Zuo, X.; Zhao, J. Extracting events and their relations from texts: A survey on recent research progress and challenges. *AI Open* **2020**, *1*, 22–39. [\[CrossRef\]](#)
2. Xi, X.; Ye, W.; Zhang, T.; Wang, Q.; Zhang, S.; Jiang, H.; Wu, W. Improving event detection by exploiting label hierarchy. In Proceedings of the ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 7688–7692.
3. Chen, Y.; Chen, T.; Van Durme, B. Joint modeling of arguments for event understanding. In Proceedings of the First Workshop on Computational Approaches to Discourse, online, 17 June 2020; pp. 96–101.
4. Bosselut, A.; Choi, Y. Dynamic knowledge graph construction for zero-shot commonsense question answering. *arXiv* **2019**, arXiv:1911.03876.
5. Gao, L.; Wu, J.; Qiao, Z.; Zhou, C.; Yang, H.; Hu, Y. Collaborative social group influence for event recommendation. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Indianapolis, IN, USA, 24–28 October 2016; pp. 1941–1944.
6. Boyd-Graber, J.; Börschinger, B. What question answering can learn from trivia nerds. *arXiv* **2019**, arXiv:1910.14464.
7. Ebner, S.; Xia, P.; Culkin, R.; Rawlins, K.; Van Durme, B. Multi-sentence argument linking. *arXiv* **2019**, arXiv:1911.03766.
8. Chen, Y.; Xu, L.; Liu, K.; Zeng, D.; Zhao, J. 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), Beijing, China, 26–31 July 2015; pp. 167–176.
9. Nguyen, T.H.; Cho, K.; Grishman, R. 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, San Diego, CA, USA, 13–15 June 2016; pp. 300–309.
10. Liu, X.; Luo, Z.; Huang, H. Jointly multiple events extraction via attention-based graph information aggregation. *arXiv* **2018**, arXiv:1809.09078.
11. Lin, Y.; Ji, H.; Huang, F.; Wu, L. A joint neural model for information extraction with global features. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7999–8009.
12. Liu, J.; Chen, Y.; Xu, J. 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, Online, 7–11 November 2021; pp. 2716–2725.
13. Du, X.; Cardie, C. Event extraction by answering (almost) natural questions. *arXiv* **2020**, arXiv:2004.13625.
14. Li, F.; Peng, W.; Chen, Y.; Wang, Q.; Pan, L.; Lyu, Y.; Zhu, Y. Event extraction as multi-turn question answering. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 829–838.
15. Lu, Y.; Lin, H.; Xu, J.; Han, X.; Tang, J.; Li, A.; Sun, L.; Liao, M.; Chen, S. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. *arXiv* **2021**, arXiv:2106.09232.
16. Li, S.; Ji, H.; Han, J. Document-level event argument extraction by conditional generation. *arXiv* **2021**, arXiv:2104.05919.
17. Grishman, R.; Sundheim, B.M. Message understanding conference-6: A brief history. In Proceedings of the COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics, Copenhagen, Denmark, 5–9 August 1996.
18. Liu, S.; Chen, Y.; He, S.; Liu, K.; Zhao, J. Leveraging framenet to improve automatic event detection. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 2134–2143.
19. Patwardhan, S.; Riloff, E. 56-A unified model of phrasal and sentential evidence for information extraction. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–7 August 2009; pp. 151–160.
20. Liao, S.; Grishman, R. Using document level cross-event inference to improve event extraction. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 11–16 July 2010; pp. 789–797.
21. Huang, R.; Riloff, E. Modeling textual cohesion for event extraction. In Proceedings of the AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012; Volume 26, pp. 1664–1670.
22. Li, Q.; Ji, H.; Huang, L. 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), Sofia, Bulgaria, 4–9 August 2013; pp. 73–82.

23. Nguyen, T.H.; Grishman, R. Event detection and domain adaptation with 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 2: Short Papers), Beijing, China, 26–31 July 2015; pp. 365–371.
24. Sha, L.; Qian, F.; Chang, B.; Sui, Z. Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
25. Feng, X.; Huang, L.; Tang, D.; Ji, H.; Qin, B.; Liu, T. A language-independent neural network for event detection. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Berlin, Germany, 7–12 August 2016; pp. 66–71.
26. Huang, L.; Ji, H.; Cho, K.; Voss, C.R. Zero-shot transfer learning for event extraction. *arXiv* **2017**, arXiv:1707.01066.
27. Zhang, T.; Whitehead, S.; Zhang, H.; Li, H.; Ellis, J.; Huang, L.; Liu, W.; Ji, H.; Chang, S.F. Improving event extraction via multimodal integration. In Proceedings of the 25th ACM International Conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 270–278.
28. Chen, Y.; Liu, S.; Zhang, X.; Liu, K.; Zhao, J. Automatically labeled data generation for large scale event extraction. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July– 4 August 2017; pp. 409–419.
29. Zhu, Z.; Li, S.; Zhou, G.; Xia, R. Bilingual event extraction: A case study on trigger type determination. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Baltimore, MD, USA, 22–27 June 2014; pp. 842–847.
30. Liu, S.; Chen, Y.; Liu, K.; Zhao, J. Exploiting argument information to improve event detection via supervised attention mechanisms. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July– 4 August 2017; pp. 1789–1798.
31. Lou, D.; Liao, Z.; Deng, S.; Zhang, N.; Chen, H. MLBiNet: A cross-sentence collective event detection network. *arXiv* **2021**, arXiv:2105.09458.
32. Yang, S.; Feng, D.; Qiao, L.; Kan, Z.; Li, D. Exploring pre-trained language models for event extraction and generation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5284–5294.
33. Wadden, D.; Wennberg, U.; Luan, Y.; Hajishirzi, H. Entity, relation, and event extraction with contextualized span representations. *arXiv* **2019**, arXiv:1909.03546.
34. Du, X.; Cardie, C. Document-level event role filler extraction using multi-granularity contextualized encoding. *arXiv* **2020**, arXiv:2005.06579.
35. Xu, R.; Liu, T.; Li, L.; Chang, B. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. *arXiv* **2021**, arXiv:2105.14924.
36. Tao, J.; Pan, Y.; Li, X.; Hu, B.; Peng, W.; Han, C.; Wang, X. Multi-Role Event Argument Extraction as Machine Reading Comprehension with Argument Match Optimization. In Proceedings of the ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 6347–6351.
37. Wang, X.D.; Leser, U.; Weber, L. BEEDS: Large-Scale Biomedical Event Extraction using Distant Supervision and Question Answering. In Proceedings of the 21st Workshop on Biomedical Language Processing, Dublin, Ireland, 26 May 2022; pp. 298–309.
38. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J.; et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
39. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv* **2019**, arXiv:1910.13461.
40. Paolini, G.; Athiwaratkun, B.; Krone, J.; Ma, J.; Achille, A.; Anubhai, R.; Santos, C.N.d.; Xiang, B.; Soatto, S. Structured prediction as translation between augmented natural languages. *arXiv* **2021**, arXiv:2101.05779.
41. Schick, T.; Schütze, H. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv* **2020**, arXiv:2001.07676.
42. Han, X.; Zhao, W.; Ding, N.; Liu, Z.; Sun, M. Ptr: Prompt tuning with rules for text classification. *AI Open* **2022**, *3*, 182–192. [[CrossRef](#)]
43. Hsu, I.; Huang, K.H.; Boschee, E.; Miller, S.; Natarajan, P.; Chang, K.W.; Peng, N. Event extraction as natural language generation. *arXiv* **2021**, arXiv:2108.12724.
44. Ye, H.; Zhang, N.; Bi, Z.; Deng, S.; Tan, C.; Chen, H.; Huang, F.; Chen, H. Learning to Ask for Data-Efficient Event Argument Extraction (Student Abstract). *Argument* **2022**, *80*, 100. [[CrossRef](#)]
45. Ding, N.; Chen, Y.; Han, X.; Xu, G.; Xie, P.; Zheng, H.T.; Liu, Z.; Li, J.; Kim, H.G. Prompt-learning for fine-grained entity typing. *arXiv* **2021**, arXiv:2108.10604.
46. Zuo, X.; Cao, P.; Chen, Y.; Liu, K.; Zhao, J.; Peng, W.; Chen, Y. LearnDA: Learnable knowledge-guided data augmentation for event causality identification. *arXiv* **2021**, arXiv:2106.01649.
47. Shin, T.; Razeghi, Y.; Logan IV, R.L.; Wallace, E.; Singh, S. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv* **2020**, arXiv:2010.15980.
48. Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; Tang, J. GPT understands, too. *arXiv* **2021**, arXiv:2103.10385.

49. Li, X.L.; Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv* **2021**, arXiv:2101.00190.
50. Hu, S.; Ding, N.; Wang, H.; Liu, Z.; Li, J.; Sun, M. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *arXiv* **2021**, arXiv:2108.02035.
51. Zhang, N.; Deng, S.; Cheng, X.; Chen, X.; Zhang, Y.; Zhang, W.; Chen, H.; Center, H.I. Drop Redundant, Shrink Irrelevant: Selective Knowledge Injection for Language Pretraining. In Proceedings of the IJCAI, Virtual, 19–27 August 2021; pp. 4007–4014.
52. Gao, L.; Choubey, P.K.; Huang, R. 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 Papers), Minneapolis, MN, USA, 2–7 June 2019.
53. Speer, R.; Chin, J.; Havasi, C.C. 5.5: An open multilingual graph of general knowledge. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (December 2016), Francisco, CA, USA, 4–9 February 2017; pp. 4444–4451.
54. Ye, H.; Zhang, N.; Deng, S.; Chen, X.; Chen, H.; Xiong, F.; Chen, X.; Chen, H. Ontology-enhanced Prompt-tuning for Few-shot Learning. In Proceedings of the ACM Web Conference 2022, Lyon, France, 25–29 April 2022; pp. 778–787.
55. Ma, Y.; Wang, Z.; Cao, Y.; Li, M.; Chen, M.; Wang, K.; Shao, J. Prompt for Extraction? PAIE: Prompting Argument Interaction for Event Argument Extraction. *arXiv* **2022**, arXiv:2202.12109.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.