CUP: Curriculum Learning based Prompt Tuning for Implicit Event Argument Extraction

Jiaju Lin¹, Qin Chen¹*, Jie Zhou², Jian Jin¹ and Liang He¹

¹School of Computer Science and Technology, East China Normal University
²School of Computer Science, Fudan University

jiaju_lin@stu.ecnu.edu.cn, {qchen, jjin, lhe}@cs.ecnu.edu.cn, jie_zhou@fudan.edu.cn

Abstract

Implicit event argument extraction (EAE) aims to identify arguments that could scatter over the document. Most previous work focuses on learning the direct relations between arguments and the given trigger, while the implicit relations with long-range dependency are not well studied. Moreover, recent neural network based approaches rely on a large amount of labeled data for training, which is unavailable due to the high labelling cost. In this paper, we propose a Curriculum learning based Prompt tuning (CUP) approach, which resolves implicit EAE by four learning stages. The stages are defined according to the relations with the trigger node in a semantic graph, which well captures the long-range dependency between arguments and the trigger. In addition, we integrate a prompt-based encoder-decoder model to elicit related knowledge from pre-trained language models (PLMs) in each stage, where the prompt templates are adapted with the learning progress to enhance the reasoning for arguments. Experimental results on two well-known benchmark datasets show the great advantages of our proposed approach. In particular, we outperform the state-of-the-art models in both fully-supervised and low-data scenarios.

1 Introduction

Event argument extraction (EAE) plays an important role in artificial intelligence, which has been widely used in global crisis monitoring and decision making [Zhan and Jiang, 2019; Mostafazadeh Davani et al., 2019; Chen et al., 2020]. Unlike traditional EAE that mainly focuses on extracting arguments within a single sentence given a trigger, implicit EAE focuses on document-level argument extraction that arguments may span over multiple sentences, which is more challenging [Ebner et al., 2020]. As shown in Figure 1, the word ‘shooting’ triggers a Conflict/attack/firearms attack event. Implicit EAE aims to extract arguments corresponding to their roles beyond the sentence level, namely Tatarstan (place), mass murder (target), Andrey Shpagonoy (attacker), and firearms (instrument).

One key challenge in implicit EAE is how to capture the long-range dependency between arguments and the given trigger. Previous methods mainly rely on the pair-wise learning paradigm that models the direct relation between each argument and trigger. Ebner et al. [2020] determined the best argument span according to the matching score between the candidate span and the trigger-role representations. Du and Cardie; Li et al.; Liu et al. [2020b; 2020; 2020] devised the trigger and role-specific questions to locate the argument spans, by treating it as Machine Reading Comprehension (MRC). Recent approaches tend to incorporate longer dependency to enhance argument extraction. Du and Cardie; Du et al. [2020a; 2020] introduced a generative transformer-based framework, which encoded the document-level context and decoded arguments one-by-one. Whereas, the order of argument generation is difficult to determine, which is vulnerable to the error propagation as shown in previous studies [Xiangyu et al., 2021].

Another challenge is how to obtain good performance in low-data scenarios. Due to the long-range complex dependency, it is usually expensive to obtain large-scale labeled data for training. Even the performance of the state-of-the-art models drops sharply when the argument is two-sentence away from the trigger [Zhang et al., 2020]. To alleviate this problem, Liu et al. [2021a] proposed implicit and ex-

Figure 1: An example of implicit EAE from RAMS. Solid lines link the trigger and [argument role]. Dash lines link the coreference entities across sentence boundaries.
plicit data augmentation methods based on MRC. Whereas, these methods rely on the quality of external data, where the domain shift or noise will have side effects on the performance. Recently, prompt-based learning has been investigated to elicit knowledge from pre-trained language models (PLMs) for low-data scenarios [Li and Liang, 2021; Lester et al., 2021]. Li et al. [2021] incorporated a template with masked arguments into the encoder-decoder framework, which conditionally generated the complete template by argument filling, and achieved good performance in the zero-shot scenarios. Whereas, the template is defined based on the event ontology that is not available in many cases. Moreover, the template text description can not well reflect the complex dependency between multiple arguments and triggers. Thus, more effective prompt templates for modeling the complex dependencies remain to be studied.

To resolve the above two challenges, we propose a Curriculum learning based Prompt tuning (CUP) approach for implicit EAE in this paper. Specifically, we first parse the document as an Abstract Meaning Representation (AMR) graph [Banarescu et al., 2013], which can well capture the long-range dependency by an explicit path from arguments to the given trigger. To obtain an effective order for argument extraction, we present a curriculum learning framework that learns to extract arguments by four stages with increasing difficulties, inspired by the cognitive process of learning from easy-level to hard-level gradually. The four learning stages are Center-based Extraction, Neighborhood-based Extraction, Document-based Extraction, and Document-based Extraction without Clues. The learning difficulty is defined by the hops between the extraction part and the trigger node in the AMR graph.

In each stage, we utilize a prompt-based encoder-decoder model, which alleviates the data insufficiency problem by eliciting related knowledge from PLMs. It is notable that the prompt templates are adapted according to the learning difficulties in different stages. For the first two stages that the sentences are near the trigger nodes (easy-level), we utilize the textual templates from annotation guidelines. While for the last two stages that focus on document extraction (hard-level), we additionally devise various graph-based templates, which enhance the reasoning ability across multiple arguments and triggers. For inference in testing, we present a pipeline decoding method, which decodes the arguments gradually from the trigger around sentences to the whole document.

Experiments are performed on two well-known benchmark datasets, namely RAMS [Ebner et al., 2020] and WikiEvents [Li et al., 2021], and the results show the effectiveness of our proposed approach. Particularly, we outperform the state-of-the-art approaches in fully-supervised and low-data scenarios. The main contributions of our work are as follows:

- To the best of our knowledge, it is the first attempt to perform implicit EAE with four explicit curriculum learning stages, which well capture the long-range dependency and extraction order.
- Different templates are adapted for prompt tuning in different learning stages, which can further enhance the reasoning ability of the target argument in low-data scenarios.
- We conduct elaborate analyses of the experimental results on two benchmark datasets, and provide a better understanding of the effectiveness of our approach.

2 Our Approach

The framework of our proposed approach is illustrated in Figure 2. Specifically, we first construct an AMR graph for the document that describes the event, which well reflects the long-range dependency between arguments and the given trigger. Then, different parts of the document are selected according to the distance with the trigger node in the AMR graph, which are then used for implicit EAE by four-staged curriculum learning. In particular, we integrate different prompt templates into the pre-trained language model (BART) [Lewis et al., 2020] in different stages, which generates the complete template by argument prediction. During inference, we utilize a pipeline decoder, which also generates the arguments gradually according to the learning progress as shown in Figure 3.

2.1 AMR Graph Construction

Abstract Meaning Representation (AMR) graph [Banarescu et al., 2013] is a sembanking language that represents the semantic structure of a sentence. Compared to traditional dependency parsing, the nodes in AMR are entities instead of words, and the relations are more logical and less vulnerable to the syntactic representation or word order variations. Therefore, we utilize AMR to model the interdependency between arguments and the given trigger. First, we utilize a transition-based AMR parser [Zhou et al., 2021] trained on AMR 2.0 annotations to generate an AMR graph for each sentence in a document. Considering that some entities will be split into multiple tokens and represented by several nodes, we merge them into a single node to obtain a complete representation. Then, we link sentence-level AMR graphs together by coreferenced entities shared in the document. Here we employ ready-made coreference resolver provided by AllenNLP\(^1\) [Lee et al., 2018]. Finally, a large AMR graph corresponding to the document is obtained. Formally, give a document $D$, its document-level AMR graph is represented as $G_D = (V_D, E_D)$, where $V_D$ represents the nodes including arguments and the given trigger, and $E_D$ indicates the edges that reflect the dependency between nodes.

2.2 Curriculum Learning with Prompts

Motivated by the cognitive process of humans and animals that learn knowledge from easy to hard gradually, we propose a curriculum learning [Bengio et al., 2009] framework to resolve implicit EAE by several stages, which learns to extract arguments from trigger around sentences to the whole document. Specifically, the learning stages are defined according to the number of hops from the given trigger to a target argument in the document AMR graph, which well reflects the dependence length and extraction difficulties. Four stages are

\(^1\)https://demo.allennlp.org/coreference-resolution
Figure 2: Overview of curriculum learning with prompts during training. The training stages go on from left to right. In the first two stages, we select sentences containing or surrounding the trigger word. The prompts are guideline-based, gradually filled with clues. In the last two stages, the input are expanded to the full document, with additional structualized prompts generated from the AMR graph.

Figure 3: Pipeline decoding during inference.

explicitly presented, namely Center-based Extraction (Cent Ex), Neighborhood-based Extraction (Neigh Ex), Document-based Extraction (Doc Ex), and Document-based Extraction without Clues (Doc Ex (w/o Cl)). Moreover, we incorporate different prompt-tuning strategies in different learning stages, which can help elicit knowledge from PLMs to boost performance in low-data scenarios. The details of each learning stage are presented as follows.

Center-based Extraction. Intuitively, the sentence that contains the trigger node is more pertinent to the event, which can be deemed as the centered sentence. In the first stage, we focus on extracting arguments from the centered sentence (i.e., the sentence with orange background in Figure 2). In other words, we only need to conduct sentence-level EAE and extract the nearest arguments with the shortest dependency on the trigger in this stage. A pre-trained language model, namely BART [Lewis et al., 2020], is used for argument extraction. Specifically, for a centered sentence $x_i$ with trigger inside, we first use special token \( \langle \text{trg} \rangle \) to markup the position of the trigger word and get $x_i$ as:

$$x_i = [t_0; \ldots; t_{\text{trg}-1}; \langle \text{trg} \rangle; t_{\text{trg}}; \langle \text{trg} \rangle; t_{\text{trg}+1}; \ldots; t_l] \quad (1)$$

where $t_i$ indicates the $i$th token in the sentence, and $\text{trg}$ is the index of the trigger word. Then, $\pi_i$ is wrapped along with a prompt template $p$ by a function $\lambda (\cdot, \cdot)$:

$$X_p = \lambda (p, \pi_i) = \langle s \rangle p \langle /s \rangle \pi_i \langle /s \rangle \quad (2)$$

Here, $p$ is manually adapted from event annotation guidelines where all arguments are replaced by a placeholder token $\langle \text{arg} \rangle$, $\langle s \rangle$ and $\langle /s \rangle$ indicate the start and end of a sentence in BART respectively. After that, we obtain the input of the encoder, i.e., $X_p$, which is then encoded to generate a filled template that replaces the placeholder tokens by the extracted arguments.

Neighborhood-based Extraction. After extraction from the centered sentence, we expand the neighborhood sentences in which the concept nodes are near the trigger and extracted arguments in the AMR graph. As shown in Figure 2, the linkages from arguments in the neighborhood sentences (with green background) to the trigger node go through at least one coreference node, which increases the extraction difficulty. To facilitate extraction, we adapt the manual prompt template by filling placeholders with previously extracted arguments as clues, which is denoted as $p^c$. Hence, our model can capture relations among extracted arguments and infer cross-sentence arguments more efficiently. Formally, the input of the encoder is represented as:

$$X_{p^c} = \lambda (p^c, [x_{i-1}; \pi_i; x_{i+1}]) \quad (3)$$

where $x_{i-1}$ and $x_{i+1}$ represent the neighborhood sentences, and ‘;’ denotes concatenation.

Document-based Extraction. In this stage, we expand the extraction range to the whole document. Considering the arguments could span over multiple sentences, we devise a
graph-based structuralized prompt in addition to the above non-structuralized textual prompt, which aims to enhance the reasoning ability for the arguments that have long-range dependency on the trigger. To be specific, we first find the smallest subgraph of $G^D$ that contains the trigger $(\text{trg})$, the nearest argument in the centered sentence $(a_c)$, and the target argument to be extracted $(a_t)$. Then, the subgraph is transformed into a sequence by a linearization function to form the prompt template, which is formulated as:

$$p'_{g} = \text{Linear}(\text{subgraph}(G^D, \text{trg}, a_c, a_t))$$

(4)

where \text{subgraph} is a function that locates the smallest subgraph from $G^D$ which contains the nodes corresponding to $\text{trg}$, $a_c$ and $a_t$, \text{Linear} is a function that transforms the subgraph into a sequence, which can be implemented by the graph traversing algorithm as demonstrated in [Bevilacqua et al., 2021]. Here, we apply the depth-first searching (DFS) algorithm to obtain the sequence.

With the graph-based prompt $p'_{g}$ and previous text-based prompt $p^e$ (with extracted clues), we combine them with the document by the function $\lambda(\cdot, \cdot)$ to obtain two prompt-based representations as the input of the encoder respectively:

$$X_{p^e} = \lambda(p^e, D) \quad X_{p'_{g}} = \lambda(p'_{g}, D)$$

(5)

where $D$ is adapted from $D$ by surrounding the trigger with special ‘(trg)’ tokens.

Document-based Extraction without Clues. The last stage is document extraction without using previously extracted arguments as clues, which is the hardest among all stages. In addition to the raw manual prompt used in the first stage (i.e., $p$), we devise an additional structuralized prompt without any previous clues. Specifically, the shortest path between the trigger node and target argument node in the AMR graph is used to form the structuralized prompt without clues:

$$p_g = \text{path}(G^D, \text{trg}, a_t)$$

(6)

where $\text{path}$ is a function to find the shortest path from the trigger $\text{trg}$ to the target argument $a_t$ in the AMR graph $G^D$.

In this stage, we also have two varieties of inputs for the encoder, which aims to enhance document-based argument extraction with original textual prompt and structuralized path based prompt:

$$X_p = \lambda(p, D) \quad X_{p_g} = \lambda(p_g, D)$$

(7)

2.3 Training Objective

For each stage, we feed the related text with prompt, denoted as $X$, into the BART encoder, and obtain the conditional generation probability for each output token $o_c$ by decoder as:

$$p(o_c|o_{<c}, X) = \text{BART}(X)$$

(8)

The model parameters are optimized by minimizing the negative loglikelihood over the prompt with gold arguments:

$$\mathcal{L}(X) = -\sum_{c=1}^{C} \log(p(o_c|o_{<c}, X))$$

(9)

where $C$ denotes the length of the prompt template.

For the last two training stages, since there are two varieties of input with different prompts, we calculate the decoding loss for each input separately, and introduce a hyperparameter $\alpha$ to combine them as follows:

$$\mathcal{L} = \mathcal{L}(X_p) + \alpha \mathcal{L}(X_{p_g})$$

(10)

2.4 Pipeline Decoding

Most previous methods decode arguments according to their orders in the document or guideline [Du et al., 2020; Li et al., 2021], which may violate the cognitive process of learning from easy-level to hard-level [Bengio et al., 2009]. In contrast, we present a pipeline decoding for implicit EAE during inference, which is consistent with the learning process in the training period. Specifically, our pipeline decoding consists of three stages as shown in Figure 3, namely Center-based Decoding (Cent Dec), Neighborhood-based Decoding (Neigh Dec), and Document-based Decoding (Doc Dec), which decode the prompt template by filling placeholders with arguments from the centered sentence, neighborhood sentences and full document gradually. Notably, only the textual prompts based on guidelines are used in all stages since the graph-based prompts are unavailable due to the unlabelled target argument in the test data. All previous extracted arguments are used as clues to boost further decoding.

3 Experimental Setup

Datasets. We conduct extensive experiments on two implicit EAE datasets, namely RAMS [Ebner et al., 2020] and WikiEvents [Li et al., 2021], which have been widely used in previous studies [Li et al., 2021; Liu et al., 2021a]. The RAMS dataset includes 3,993 paragraphs, annotated with 139 event types and 65 semantic roles. In the WikiEvents dataset, 246 documents are collected and annotated with 50 event types and 59 semantic roles. Our experiments are conducted under informative argument setting\(^2\). We follow the official data split of each dataset. Detailed statistics are listed in Table 1. The performance is evaluated by Precision (P), Recall (R), and F1 score (F1), based on the Exact Match (EM) criterion: only when the prediction matches the gold one exactly, it is deemed as a correct prediction.

Implementations. Considering that most previous studies are based on the base-scale models (e.g., BERT-base), we employ a vanilla BART-base model in our approach for fairness of comparison. To adapt to the input length of the BART model, we truncate the documents in WikiEvents that are much longer than those in RAMS, and select the trigger surrounding sections with at most nine sentences for experiments. The experiments are conducted on a single Nividia GeForce RTX 3090. For the four stages of curriculum learning, the learning rate decreases as the learning process goes on, which is set as 1e-4, 5e-5, 3e-5 and 2e-5 respectively. The best batch size is searched from the set of $\{16, 32\}$ for each stage. The hyper-parameter of $\alpha$ that balances the two losses in Formula (10) ranges from 0 to 1 with a step of 0.1, and we

\(^2\)In this setting, the most informative mention of the argument is treated as the ground truth, which is more challenging.
set it to 0.7 as a reliable setting. All other parameters are optimized by the Adam [Kingma and Ba, 2014] algorithm. The optimal parameters are obtained based on the development set, then used for evaluation on test set.3

**Baselines.** We compare our approach with the recent advanced baselines: 1) SpanSel [Ebner et al., 2020] is a method based on span ranking, which enumerates each possible span in a document to identify the argument; 2) Head-Expand [Li et al., 2019] extends SpanSel by first identifying an argument’s head and then its region; 3) BART-Gen [Ebner et al., 2020] treats implicit EAE as a generation task, and generates the arguments based on BART architecture4; 4) DocMRC [Liu et al., 2021a] models implicit EAE as a MRC problem, which devises the role-specific questions for argument extraction; 5) DocMRC (IDA) is an enhanced version of DocMRC, which uses implicit data augmentation by pre-training on other related tasks, including question answering, natural language inference and sentence level event argument extraction.

### Results and Analyses

#### 4.1 Overall Performance

Table 2 shows the performance of our approach and the state-of-the-art baselines on RAMS and WikiEvents. For fairness of comparison, all the results are under the settings that the event types are known. We have the following observations. **First,** we achieve the best performance on the two datasets in most cases, indicating the superiority of our proposed approach without using any external data for pre-training or labelling. **Second,** our approach that integrates BART in our curriculum learning framework surpasses the BART-GEN baseline with a large magnitude, indicating that we can make the most potential of the BART architecture by curriculum learning based prompt tuning. **Third,** compared with the best baseline DocMRC (IDA) that relies on external data for augmentation, our approach outperforms it with an improvement up to 9.47% in terms of F1 on WikiEvents, which indicates the effectiveness of our prompt tuning strategies in alleviating the data insufficiency problem.

#### 4.2 Ablation Studies

We conduct ablation studies to investigate the effectiveness of each component of our approach, and the results are shown in Table 3. First, we remove one of the four curriculum learning stages each time, noted as - Cent Ex, - Neigh Ex, - Doc Ex, - Doc Ex (w/o Cl). Then, we verify the effectiveness of graph-based prompt, illustrated by - p$. For pipeline decoding, we only remove Center-based Decoding (- Cent Dec) and Neighborhood-based Decoding (- Neigh Dec) respectively, since Document-based Decoding can not be removed for complete argument extraction.

We can observe that each curriculum learning stage can help boost the performance of implicit EAE. To be specific, without Center-based Extraction, the performance on both RAMS and WikiEvents drops dramatically by about 2 points in terms of F1. This justifies that argument extraction from the centered sentence that contains the trigger is indispensable. Similar observations can be found in other stages, indicating the effectiveness of each stage for enhancing argument extraction. In particular, the last stage (Doc Ex (w/o Cl)) also plays an important role as the third stage (Doc Ex), though the only difference lies in that it does not use the extracted arguments as clues for extraction. This finding verifies the necessity of using the last stage as the hardest learning task to boost implicit EAE. Regarding prompt tuning, it is observed the graph-based prompt p brings an improvement up to 6.04% on WikiEvents with respect to F1, demonstrating the superiority of graph-based prompt for argument reasoning over long-range dependency. For pipeline decoding, we have similar observations. Each decoding step is effective, and decoding arguments from the centered sentence (Cent Dec) is also more important than other steps.

#### 4.3 Impact of Trigger-Argument Distance

To investigate whether our approach can well capture the long-range dependency between the trigger and arguments, we use the sentence-level distance to roughly model the dependency. Each sample in RAMS contains five sentences with explicit sentence boundaries. While for WikiEvents, in many cases, it splits sentences inaccurately. Thus, we split...
the document into sentences with spaCy\textsuperscript{5}, and select a five-sentence paragraph around the trigger for experiments. Table 4 illustrates the impact of sentence-level distance from triggers to arguments. We compare with two state-of-the-art baselines shown in Table 2. Note that DocMRC (IDA) additionally uses external data for training, which is excluded here for a fair comparison. We observe that our approach outperforms baselines in most cases with various distances. It is also interesting to find that the performance of all the models does not always drop when the distance increases, especially for the WikiEvents dataset. This may be related to the various syntax and discourse structures in event descriptions, and the sentence-level distance cannot completely reflect the complex dependency between arguments and the trigger. Overall, our approach performs better than the baselines even when the trigger-argument distance increases to two sentences (Table 4) or the whole document (Table 2).

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
Method & Trigger-Argument Distance on RAMS & \\
 & -2[4%] & -1[8%] & 0[83%] & 1[4%] & 2[2%] \\
\hline
BART-Gen & 17.7 & 16.8 & 44.8 & 16.6 & 9.0 \\
DocMRC & 15.2 & 12.6 & 48.6 & 20.5 & 3.4 \\
CUP (Ours) & \textbf{19.3} & \textbf{22.2} & \textbf{49.6} & 17.4 & 8.6 \\
\hline
Method & Trigger-Argument Distance on WikiEvents & \\
 & -2[8%] & -1[10%] & 0[87%] & 1[8%] & 2[7%] \\
\hline
BART-Gen & 11.1 & 6.9 & 43.2 & 2.9 & 7.7 \\
DocMRC & 13.9 & 5.6 & 44.4 & \textbf{7.3} & 8.0 \\
CUP (Ours) & \textbf{15.7} & \textbf{7.7} & \textbf{51.4} & 6.7 & \textbf{8.9} \\
\hline
\end{tabular}
\caption{Impact of the trigger-argument distance. The results marked with * is from [Liu et al., 2021a]. Other results are reproduced due to the different experimental settings, such as sentence split method or training data.}
\end{table}

4.4 Performance in Low-data Scenarios

To examine the performance under low-data scenarios, we vary the ratio of the training data from 1% to 30%, and the results are shown in Figure 4. It is observed that our approach outperforms the two state-of-the-art baselines on both datasets. In particular, our approach overtakes DocMRC trained by 20% of the training data, with only half of the samples (10%) on RAMS dataset. All these observations verify the effectiveness of our prompt tuning strategies that elicit related knowledge from pre-trained language models, which can help alleviate the data insufficiency problem.

5 Related Work

Event Argument Extraction (EAE) aims to identify arguments from text with a given trigger. Sentence-level EAE has achieved great progress in previous studies [Chen et al., 2015; Liu et al., 2018]. While implicit EAE that arguments could span over multiple sentences is not well studied. Previous works treat it as a semantic role labeling task [Zhang et al., 2020]. Recently, there is a trend to formulate implicit EAE as Machine Reading Comprehension (MRC) [Du and Cardie, 2020b], where manual questions are designed to locate the answer as the target argument. Whereas, the MRC-based methods focus on learning the pairwise trigger-argument relations, which neglect the long-range dependency between trigger and arguments. In addition, it is difficult to obtain the training data due to the expensive cost. [Liu et al., 2021a] attempted to alleviate the data sparsity problem by implicit and explicit data augmentations. However, these methods rely on external data, which is vulnerable to domain shift or noise.

Prompt Tuning is introduced as a new paradigm of natural language processing [Liu et al., 2021b]. Rather than introducing a classifier during fine-tuning, prompt-based methods wrap input context with a continuous or discrete vector with unfilled slots called prompt. Specifically, prompt engineering is a key step for prompt tuning, which creates a proper prompt template for the task. Petroni et al. [2019] first proposed a manual designed prompt which transformed classification as a cloze filling task. Other studies attempt to explore automatic searched prompts to alleviate the manual cost [Li and Liang, 2021][Lester et al., 2021]. Due to the complexity of implicit EAE, the appropriate prompt tuning methods remain to be studied. In this work, we explore different prompt tuning strategies in different learning stages.

Curriculum Learning is a learning tactic proposed by Bengio et al. [2009], where a model is trained by training data with increasing complexity. Wei et al. [2021] introduced a curriculum knowledge distillation strategy to incorporate related arguments for implicit EAE. Whereas, the learning difficulty is not explicitly defined, and the argument relations are not included in inference. In this paper, we propose a well-defined multi-staged curriculum learning framework, which models the interdependency for both training and inference.

6 Conclusions

In this paper, we propose a curriculum learning based prompt tuning approach for implicit EAE. Extensive experiments have verified the effectiveness of each learning stage. Furthermore, our approach consistently outperforms the state-of-the-art models in both full-training or low-data scenarios, by incorporating different prompt-tuning strategies in different learning stages. In particular, our graph-based prompt significantly boosts the performance by up to 6.04% in terms of F1. In the future, we will explore more prompts to enhance reasoning for target arguments with long-range dependency.

\textsuperscript{5}https://spacy.io/
References


[Li et al., 2021] Sha Li, Heng Ji, and Jiawei Han. Document-level event argument extraction by conditional generation. In NAACL, pages 894–908, 2021.


