

Bi-Directional Iterative Prompt-Tuning for Event Argument Extraction

Lu Dai¹, Bang Wang¹, Wei Xiang¹, Yijun Mo^{2*}

¹School of Electronic Information and Communications,
Huazhong University of Science and Technology, Wuhan, China

²School of Computer Science and Technology,
Huazhong University of Science and Technology, Wuhan, China
{dailu18, wangbang, xiangwei, moyj}@hust.edu.cn

Abstract

Recently, prompt-tuning has attracted growing interests in event argument extraction (EAE). However, the existing prompt-tuning methods have not achieved satisfactory performance due to the lack of consideration of entity information. In this paper, we propose a bi-directional iterative prompt-tuning method for EAE, where the EAE task is treated as a cloze-style task to take full advantage of entity information and pre-trained language models (PLMs). Furthermore, our method explores event argument interactions by introducing the argument roles of contextual entities into prompt construction. Since template and verbalizer are two crucial components in a cloze-style prompt, we propose to utilize the role label semantic knowledge to construct a semantic verbalizer and design three kinds of templates for the EAE task. Experiments on the ACE 2005 English dataset with standard and low-resource settings show that the proposed method significantly outperforms the peer state-of-the-art methods. Our code is available at <https://github.com/HustMinsLab/BIP>.

1 Introduction

As a key step of event extraction, event argument extraction refers to identifying event arguments with predefined roles. For example, for an "Attack" event triggered by the word "fired" in the sentence "*Iraqis have **fired** sand missiles and AAA at aircraft*", EAE aims to identify that "Iraqis", "missiles", "AAA" and "aircraft" are event arguments with the "Attacker", "Instrument", "Instrument" and "Target" roles, respectively.

In order to exploit the rich linguistic knowledge contained in pre-trained language models, fine-tuning methods have been proposed for EAE. The paradigm of these methods is to use a pre-trained language model to obtain semantic representations,

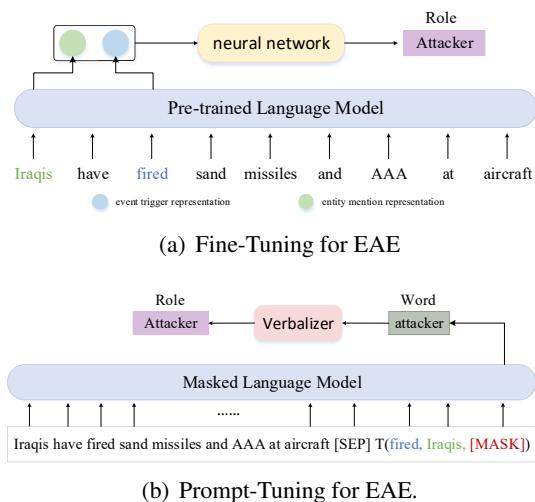


Figure 1: Illustration of fine-tuning and prompt-tuning methods for predicting the argument role of the entity mention "Iraqis" in the event triggered by the word "fired".

and then feed these representations into a well-designed neural network to extract event arguments. For example in Figure 1(a), an event trigger representation and an entity mention representation are first obtained through a pre-trained language model, and then input to a designed neural network, such as hierarchical modular network (Wang et al., 2019) and syntax-attending transformer network (Ma et al., 2020), to determine the argument role that the entity mention plays in the event triggered by the trigger. However, there is a significant gap between the EAE task and the objective form of pre-training, resulting in the poor utilization of the prior knowledge in PLMs. Additionally, fine-tuning methods heavily depend on extensive annotated data and perform poorly in low-resource data scenarios.

To bridge the gap between the EAE task and the pre-training task, prompt-tuning methods (Li et al., 2021; Ma et al., 2022; Hsu et al., 2022; Liu et al., 2022) recently have been proposed to for-

* Corresponding author: Yijun Mo

malize the EAE task into a more consistent form with the training objective of generative pre-trained language models. These methods achieve significantly better performance than fine-tuning methods in low-resource data scenarios, but not as good as the state-of-the-art fine-tuning method ONEIE (Lin et al., 2020) in high-resource data scenarios.

To achieve excellent performance in both low-resource and high-resource data scenarios, we leverage entity information to model EAE as a cloze-style task and use a masked language model to handle the task. Figure 1(b) shows a typical cloze-style prompt-tuning method for EAE. The typical prompt-tuning method suffers from two challenges: (i) The typical human-written verbalizer (Schick and Schütze, 2021) is not a good choice for EAE. The human-written verbalizer is to manually assign a label word to each argument role. For example in Figure 1(b), we choose the "attacker" as the label word of "Attacker" role. However, an argument role may have different definitions in different types of events. For example, the "Entity" role refers to "the voting agent" and "the agents who are meeting" in the "Elect" and "MEET" events, respectively. (ii) Event argument interactions are not explored. Existing work (Sha et al., 2018; Xiangyu et al., 2021; Ma et al., 2022) has demonstrated the usefulness of event argument interactions for EAE. For the "Attack" event triggered by the word "fired" in Figure 1, given that "missiles" is an "Instrument", it is more likely to correctly classify "AAA" into the "Instrument" role.

In this paper, we propose a *bi-directional iterative prompt-tuning* (BIP) method to alleviate the aforementioned challenges. To capture argument interactions, a forward iterative prompt and a backward iterative prompt are constructed to utilize the argument roles of contextual entities to predict the current entity's role. For the verbalizer, we redefine the argument role types and assign a virtual label word to each argument role, where the initial representation of each virtual label word is generated based on the semantic of the argument role. In addition, we design three kind of templates: hard template, soft template, and hard-soft template, which are further discussed in the experimental section. Extensive experiments on the ACE 2005 English dataset show that the proposed method can achieve the state-of-the-art performance in both low-resource and high-resource data scenarios.

2 Related Work

In this section, we review the deep learning methods for event argument extraction and prompt-tuning methods for natural language processing.

2.1 Event Argument Extraction

Early deep learning methods use various neural networks to capture the dependencies in between event triggers and event arguments to extract event arguments, such as convolutional neural network (CNN)-based models (Chen et al., 2015), recurrent neural network (RNN)-based models (Nguyen et al., 2016; Sha et al., 2018) and graph neural networks (GNN)-based models (Liu et al., 2018; Dai et al., 2021). As pre-trained language models have been proven to be powerful in language understanding and generation (Devlin et al., 2019; Liu et al., 2019; Lewis et al., 2020), some PLM-based methods have been proposed to extract event arguments. These methods can be divided into two categories: fine-tuning and prompt-tuning ones.

Fine-tuning methods aim to design a variety of neural network models to transfer pre-trained language models to EAE task. According to the modeling manner of EAE task, existing fine-tuning work can be further divided into three groups: classification-based methods (Wang et al., 2019; Wadden et al., 2019; Lin et al., 2020; Ma et al., 2020; Xiangyu et al., 2021); machine reading comprehension-based methods (Du and Cardie, 2020; Li et al., 2020; Liu et al., 2020); generation-based methods (Paolini et al., 2020; Lu et al., 2021). Prompt-tuning methods aim to design a template to provide useful prompt information for pre-trained language models to extract event arguments (Li et al., 2021; Ma et al., 2022; Hsu et al., 2022; Liu et al., 2022). For example, Li et al. (2021) create a template for each event type based on the event ontology definition and model the EAE task as the conditional text generation. This method acquires event arguments by comparing the designed template with the generated natural language text. Hsu et al. (2022) improve the method of Li et al. (2021) by replacing the non-semantic placeholder tokens in the designed template with words with role label semantics.

2.2 Prompt-tuning

The core of prompt-tuning is to transform the given downstream task into a form that is consistent with a training task of the pre-trained language

models (Liu et al., 2021). As prompt-tuning makes better use of prior knowledge contained in pre-trained language models, this new paradigm is beginning to become popular in NLP tasks and has achieved promising performance (Seoh et al., 2021; Han et al., 2021; Cui et al., 2021; Hou et al., 2022; Hu et al., 2022; Chen et al., 2022). For example, Cui et al. (2021) use candidate entity spans and entity type label words to obtain templates, and recognize entities based on the pre-trained generative language model’s score for each template. Hu et al. (2022) convert the text classification task to a masked language modeling problem by predicting the word filled in the "[MASK]" token, and propose a knowledgeable verbalizer to map the predicted word into a label. Chen et al. (2022) consider the relation extraction problem as a cloze task and use the relation label semantic knowledge to initialize the virtual label word embedding for each relation label.

3 Model

In this section, we first introduce the problem description of event argument extraction and the overall framework of our bi-directional iterative prompt-tuning method, then explain the details of designed semantical verbalizer, three different templates, and model training.

3.1 Problem Description

As the most common ACE dataset provides entity mention, entity type and entity coreference information, we use these entity information to formalize event argument extraction into the argument role prediction problem of entities. The detailed problem description is as follow: *Given a sentence S , an event trigger t with event type, and n entities $\{e_1, e_2, \dots, e_n\}$, the goal is to predict the argument role of each entity in the event triggered by t and output a set of argument roles $\{r_1, r_2, \dots, r_n\}$.*

In this paper, the argument role prediction problem is casted as a cloze-style task through a template $T(\cdot)$ and verbalizer. For the trigger t and entity e_i , a template $T(t, e_i, [\text{MASK}])$ is constructed to query the argument role that the entity e_i plays in the event triggered by t . For example in Figure 1(b), the template $T(\text{fired}, \text{Iraqis}, [\text{MASK}])$ can be set as "For the attack event triggered by the fired, the person, Iraqis, is [MASK]", where "attack" represents the event type of the trigger "fired" and "person" represents the entity type of the entity

"Iraqis". Then the input of the i -th entity e_i is:

$$x_i = S [\text{SEP}] T(t, e_i, [\text{MASK}]). \quad (1)$$

The verbalizer is a mapping from the label word space to the argument role space. Let l_j denote the label word that is mapped into the role r_j , the confidence score that the i -th entity is classified as the j -th role type is:

$$s_{ij} = C_i([\text{MASK}] = l_j), \quad (2)$$

where C_i is the output of a pre-trained masked language model at the masked position in x_i , i.e. the confidence score of each word in the dictionary filled in the [MASK] token.

3.2 Overall Framework

Figure 2 presents the overall architecture of our bi-directional iterative prompt-tuning method, consisting of a forward iterative prompt (FIP) and a backward iterative prompt (BIP). The forward iterative prompt predicts the argument role of each entity iteratively from left to right until argument roles of all entities are obtained. For example in Figure 2, the order of entities is "Iraqis \rightarrow missiles \rightarrow AAA \rightarrow aircraft".

In order to utilize the predicted argument role information to classify the current entity into the correct role, we introduce the argument roles of the first $i-1$ entities into the template of the i -th entity. The template of the i -th entity in the forward iterative prompt can be represented as:

$$FIP(e_i) = T(t, e_1, \vec{l}_1, \dots, e_{i-1}, \vec{l}_{i-1}, e_i, [\text{MASK}]), \quad (3)$$

where \vec{l}_j is the role label word of the j -th entity predicted by the forward iterative prompt. For example in Figure 2, \vec{l}_1 is the word "attacker". Then the confidence score distribution of the i -th entity over all argument roles in the forward iterative prompt can be computed by

$$\vec{s}_i = \text{MLM}(S [\text{SEP}] FIP(e_i)). \quad (4)$$

\vec{l}_i is the word corresponding to the argument role with the highest value in \vec{s}_i .

Similarly, the backward iterative prompt predicts the argument role of each entity in a right-to-left manner. The argument role confidence score distribution of the i -th entity in the backward iterative prompt can be computed by:

$$BIP(e_i) = T(t, e_n, \overleftarrow{l}_n, \dots, e_{i+1}, \overleftarrow{l}_{i+1}, e_i, [\text{MASK}]), \quad (5)$$

$$\overleftarrow{s}_i = \text{MLM}(S [\text{SEP}] BIP(e_i)). \quad (6)$$

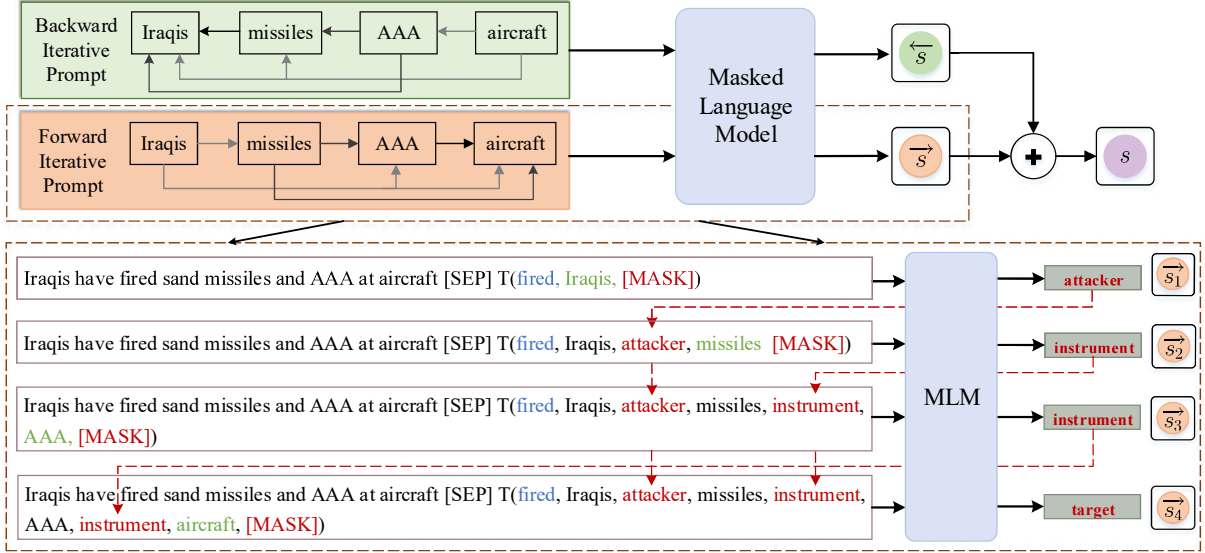


Figure 2: The overall architecture of our bi-directional iterative prompt-tuning method shown with an example predicting argument roles of "Iraqis", "missiles", "AAA", and "aircraft" in the "Attack" event triggered by "fired", where blue font represents the given trigger and green font represents the given entity.

Then we can obtain the final argument role confidence score distribution of the i -th entity by

$$s_i = \vec{s}_i + \overleftarrow{s}_i. \quad (7)$$

Finally, the argument role label with the highest score is chosen as the role prediction result.

3.3 Semantical Verbalizer

To tackle the problem that an argument role may have different definitions in different types of events, we reconstruct the set of argument role types and design a semantical verbalizer. Specifically, we further divide the argument role that participates in multiple types of events into multiple argument roles that are specific to event types. For example, the "Entity" role is divided into "Elect:Entity", "Meet:Entity", and etc. Since the "Place" role has the same meaning in all types of events, we do not consider to divide it.

For each new argument role, the semantical verbalizer constructs a virtual word to represent the role and initializes the representation of the virtual word with the semantic of the argument role. Let a m -word sequence $\{q_{i1}, q_{i2}, \dots, q_{im}\}$ denote the semantic description of the argument role r_i , the initial representation of the label word l_i that is mapped into the role r_i can be computed by:

$$\mathbf{E}(l_i) = \frac{1}{m} \sum_{j=1}^m \mathbf{E}(q_{ij}), \quad (8)$$

where \mathbf{E} is the word embedding table of a pre-trained masked language model.

For redefined argument roles, different argument roles may have the same semantics, such as "Appeal:Adjudicator" and "Sentence:Adjudicator". Therefore, it is easy to misclassify the entity with "Appeal:Adjudicator" role into the "Sentence:Adjudicator" role. In order to solve the problem, we use the event structure information to extract arguments. For an event with the "Appeal" type, its role label can only be "Appeal:Defendant", "Appeal:Adjudicator" and "Appeal:Plaintiff".

3.4 Templates

To take full advantage of event type, trigger, and entity information, the designed template should contain event types, triggers, entity types, and entity mentions. Since some entity types and event types are not human-understandable words, such as "PER" and "Phone-Write", we need to convert each entity (event) type into a human-understandable text span. For example, we use "person" and "written or telephone communication" as the text spans for "PER" and "Phone-Write" respectively.

Let $M_i = \{\varepsilon_{i1}, \varepsilon_{i2}, \dots, \varepsilon_{id}\}$ denote the entity mention set of the i -th entity, the word sequence of the i -th entity can be represented as:

$$\hat{e}_i = \varepsilon_{i1} \text{ OR } \varepsilon_{i2} \text{ OR } \dots \text{ OR } \varepsilon_{id}. \quad (9)$$

We use w^t to denote the text span of event type of the given trigger and w_i^e to denote the text span

Template Type	Template Example
Hard Template	For the <u>attack</u> event triggered by the <u>fired</u> , the person, Iraqis, is attacker, the <u>weapon</u> , <u>missiles</u> , is [MASK]
Soft Template	<u>attack</u> <u>fired</u> person Iraqis attacker <u>weapon</u> <u>missiles</u> [V1] [V2] [V3] [MASK] [V4] [V5] [V6]
Hard-Soft Template	[V1] <u>attack</u> [V2] <u>fired</u> [V3] [V4] person [V5] Iraqis [V6] attacker [V4] <u>weapon</u> [V5] <u>missiles</u> [V6] [MASK]

Figure 3: Examples of three different templates with trigger "fired" and entity "missiles", where green font represents the given trigger, green underlined font represents the text span of event type, blue font represents the given entity, blue underlined font represents the text span of entity type.

of the entity type of the i -th entity. For the given trigger t and i -th entity e_i , three different templates of forward iterative prompt are designed as follows:

- **Hard Template:** All known information are connected manually with natural language. "For the w^t event triggered by the t , the w_1^e , \hat{e}_1 , is \vec{l}_1 , ... , the w_{i-1}^e , \hat{e}_{i-1} , is \vec{l}_{i-1} , the w_i^e , \hat{e}_i , is [MASK]"
- **Soft Template:** Add a sequence of learnable pseudo tokens after all known information. " w^t t w_1^e \hat{e}_1 \vec{l}_1 ... w_{i-1}^e \hat{e}_{i-1} \vec{l}_{i-1} w_i^e \hat{e}_i [V1] [V2] [V3] [MASK] [V4] [V5] [V6]"
- **Hard-Soft Template:** All known information are connected with learnable pseudo tokens. "[V1] w^t [V2] t [V3] [V4] w_1^e [V5] \hat{e}_1 [V6] \vec{l}_1 , ... , [V4] w_{i-1}^e [V5] \hat{e}_{i-1} [V6] \vec{l}_{i-1} [V4] w_i^e [V5] \hat{e}_i [V6] [MASK]"

Pseudo tokens are represented by "[Vi]". The embedding of each pseudo token is randomly initialized and optimized during training.

3.5 Training

During training, gold argument roles are used to generate the template of each entity in forward iterative prompt and backward iterative prompt. The optimization objective is to ensure that the masked language model can predict argument role accurately in both forward iterative prompt and backward iterative prompt. We use $\vec{p}_{t,i}$ and $\overleftarrow{p}_{t,i}$ to represent the probability of the entity e_i playing each role type in the event triggered by t in forward and backward iterative prompt respectively. The loss function is defined as follows:

$$\vec{p}_{t,i} = \text{softmax}(\vec{s}_i), \quad \overleftarrow{p}_{t,i} = \text{softmax}(\overleftarrow{s}_i),$$

$$\mathbb{L} = - \sum_{t \in \mathbb{T}} \sum_{i=1}^{n_t} (\log(\vec{p}_{t,i}(\tilde{r}_{t,i})) + \log(\overleftarrow{p}_{t,i}(\tilde{r}_{t,i}))),$$
(10)

where \mathbb{T} is the event trigger set in the training set, n_t is the number of entities contained in the same sentence as the event trigger t , and $\tilde{r}_{t,i}$ is the correct argument role of the i -th entity playing in the event triggered by t .

4 Experiments

4.1 Experimental Setup

We evaluate our proposed method on the most widely used event extraction dataset, ACE 2005 English dataset¹ (Doddington et al., 2004). Following the previous work (Wadden et al., 2019; Lin et al., 2020; Ma et al., 2022), the dataset is pre-processed and divided into training/development/test set, where 33 event subtypes, 7 entity types and 22 argument roles are considered in the processed dataset. As event argument extraction task is only focused on, we use gold entities and event triggers to conduct experiments.

We use Bert-base(containing around 110 millions parameters) (Devlin et al., 2019) and Roberta-base(containing around 125 millions parameters) (Liu et al., 2019) models to predict the masked words and train each model with AdamW, where the batch size is set to 4 and the learning rate is set to $1e-5$. For the low-resource setting, we generate some subsets containing (1%, 5%, 10%, 20%, 50%, 75%) of the fulling training set in the same way as (Hsu et al., 2022). In each experiment, the masked language model is trained by a subset and evaluated by the fulling development and test sets. All experiments are run on a NVIDIA Quadro P4000 GPU.

4.2 Baselines

Two categories of state-of-the-art methods are compared with our proposed method.

¹<https://catalog.ldc.upenn.edu/LDC2006T06>

PLM	Model	Eval	Argument Identification			Role Classification		
			P	R	F	P	R	F
Bert	HMEAE (EMNLP, 2019)	SM	65.22	68.08	66.62	60.06	62.68	61.34
		FM	73.67	72.70	73.18	66.86	65.99	66.42
	ONEIE (ACL, 2020)	SM	73.65	71.72	72.67	69.31	67.49	68.39
		FM	79.48	75.77	77.58	74.89	71.39	73.09
	BERD (ACL, 2021)	SM	68.83	66.62	67.70	63.25	61.22	62.22
		FM	76.01	71.04	73.55	69.63	65.26	67.37
Roberta	HMEAE (EMNLP, 2019)	SM	70.37	69.24	69.80	64.00	62.97	63.48
		FM	76.58	72.55	74.51	69.49	65.84	67.62
	ONEIE (ACL, 2020)	SM	72.86	73.18	73.02	69.81	70.12	69.96
		FM	78.55	79.12	78.84	<u>75.22</u>	<u>75.77</u>	75.50
	BERD (ACL, 2021)	SM	69.03	69.53	69.28	63.24	63.70	63.47
		FM	75.72	73.28	74.48	69.08	66.86	67.95
Bart	DEGREE(EAE) (NAACL, 2022)	SM	70.39	68.95	69.66	65.77	64.43	65.10
		FM	<u>79.20</u>	75.60	77.37	74.16	70.80	72.44
	PAIE (ACL, 2022)	SM	72.16	71.12	71.64	68.65	66.71	67.67
		FM	76.75	<u>79.55</u>	78.13	72.82	74.22	73.51
Bert	BIP(our)		75.54	81.29	78.31	71.60	77.05	74.23
Roberta	BIP(our)		78.17 (-1.31)	86.40 (+6.85)	82.08 (+3.24)	75.26 (+0.04)	83.19 (+7.42)	79.03 (+3.53)

Table 1: Experiment results of our proposed method with hard template and baselines, where the boldface is the best results, the underline is the second best results, and results of baselines are our re-implementations. Due to the limited memory of our GPU, only base-version models are adopted to perform experiments.

Fine-tuning Methods:

- **HMEAE** (Wang et al., 2019) is a hierarchical modular model that uses the superordinate concepts of argument roles to extract event arguments.
- **ONEIE** (Lin et al., 2020) is a neural framework that leverages global features to jointly extract entities, relations, and events. When applying ONEIE to the EAE task, we also use gold entity mentions and event triggers to extract event arguments, without considering the relations.
- **BERD** (Xiangyu et al., 2021) is a bi-directional entity-level recurrent decoder that utilizes the argument roles of contextual entities to predict argument roles entity by entity.

Prompt-tuning Methods:

- **DEGREE(EAE)** (Hsu et al., 2022) summarizes an event into a sentence based on a designed prompt containing the event type, trigger, and event-type-specific template. Then event arguments can be extracted by comparing the generated sentence with the event-type-specific template.
- **PAIE** (Ma et al., 2022) is an encoder-decoder architecture, where the given context and designed event-type-specific prompt are input into the encoder and decoder separately to extract event argument spans.

4.3 Evaluation

Since we use an entity as a unit for argument role prediction, an event argument is correctly identified if the entity corresponding to the argument is predicted to be the non-None role type. The argument is further be correctly classified if the predicted role type is the same as the gold label.

For the above baselines, they consider that an event argument is correctly classified only if its offsets and role type match the golden argument, which can be called "**strict match (SM)**". In order to compare our model with baselines more fairly, we use a "**flexible match (FM)**" method to evaluate these baselines, that is, an argument is correctly classified if its offsets match any of the entity mentions co-referenced with the golden argument and role type match the golden argument.

Same as the previous work, the standard micro-averaged Precision(P), Recall(R), and F1-score(F1) are used to evaluate all methods.

4.4 Overall Results

Table 1 compares the overall results between our model and baselines, from which we have several observations and discussions.

(1) BIP(Roberta) gains the significant improvement in event argument extraction. The F1-scores of BIP(Roberta) are more than 9% higher than those of all baselines obtained by the strict match evaluation method. Even using the flexible match method to evaluate baselines, the

Model	Role Classification		
	P	R	F1
BIP(our)	75.26	83.19	79.03
BIP(forward)	76.06	78.95	77.47
BIP(backward)	75.94	76.61	76.27
-BI	78.79	76.02	77.38
-SV	74.79	78.07	76.39
-BI-SV	78.19	74.42	76.25

Table 2: An ablation study of our proposed method.

BIP(Roberta) method also outperforms the state-of-the-art ONEIE(Roberta) by 3.24% increase of F1-score in term of argument identification and 3.53% increase of F1-score in term of role classification.

(2) Comparing with the strict match, the flexible match achieves 5% to 7% F1-score improvements in term of argument identification and role classification. These results indicate that the trained argument extraction models can indeed identify the entity mention co-referenced with the golden argument as the event argument. In addition, in the actual application scenarios, we only pay attention to which entity is the event argument, not which mention in an entity is the event argument. Therefore, it is more reasonable and efficient to predict argument roles in unit of entity than entity mention.

(3) Roberta-version methods outperform Bert-version methods. In particular, for our proposed BIP method, Roberta further gains 3.77% and 4.8% F1-score improvements on argument identification task and role classification task respectively. These improvements can be explained by Roberta using a much larger training dataset than Bert and removing the next sentence prediction task. In the following experiments, we only consider Roberta-version methods.

4.5 Ablation Study

Table 2 presents an ablation study of our proposed BIP method. BIP(forward) only considers the forward iterative prompt to extract event arguments. BIP(backward) only considers the backward iterative prompt. BIP-BI does not use a bi-directional iterative strategy to consider argument interactions, i.e. predicts the argument role of each entity separately. BIP-SV replaces our designed semantical verbalizer with a human-written verbalizer, where each label word is manually selected from a pre-trained language model vocabu-

lary. BIP-BI-SV uses neither the bi-directional iterative strategy nor the semantical verbalizer. Some observations on the ablation study are as follows:

(1) Compared with the method BIP, the performance of BIP(forward) and BIP(backward) is decreased by 1.56% and 2.76% F1-score in term of role classification, respectively. These results clearly demonstrate that the bi-directional iterative prompt-tuning can further improve the performance by comparing with one direction.

(2) Comparing with the methods BIP-BI and BIP-BI-SV, the methods BIP and BIP-SV can further improve the performance of role classification in terms of 1.65% and 0.14% increase of F1-score, respectively. These results suggest that the bi-directional iterative strategy is useful for event argument extraction. In addition, we notice that the improvement brought by our bi-directional iterative strategy for the method BIP-BI is higher than BIP-BI-SV. This suggests that the more accurate the independent predicted argument role of each entity, the greater improvement the bi-directional iterative strategy will bring to the performance of argument extraction.

(2) The methods BIP and BIP-BI are respectively outperform the methods BIP-SV and BIP-BI-SV by 2.64% and 1.13% F1-score in term of role classification. These results illustrate that our semantical verbalizer is more effective than a human-written verbalizer for event argument extraction.

4.6 Low-Resource Event Argument Extraction

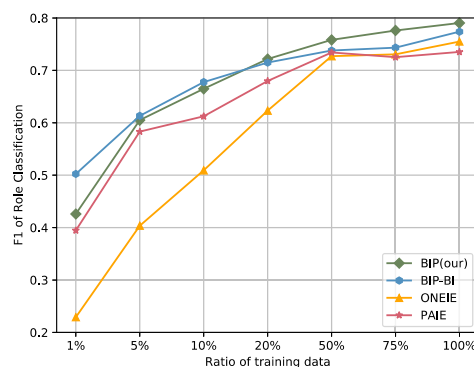


Figure 4: Performances against the ratio of training data, while ONEIE and PAIE are evaluated by flexible match.

Figure 4 presents the performance of our BIP, BIP-BI and two state-of-the-art methods in both low-resource and high-resource data scenarios. We

Sentence 1: Swapping smiles, handshakes and hugs at a joint press appearance after talks linked to Saint Petersburg’s 300th anniversary celebrations, <i>Bush</i> and <i>Putin</i> set out to recreate the buddy atmosphere of their previous encounters.					
Event Trigger: talks , Event Type: Meet					
Extraction Results:					
Entity	BIP	BIP(forward)	BIP(backward)	BIP-BI	BIP-SV
Bush	Entity(✓)	Entity(✓)	None(×)	Entity(✓)	Entity(✓)
Putin	Entity(✓)	Entity(✓)	None(×)	None(×)	Entity(✓)
Sentence 2: Earlier Saturday, <i>Baghdad</i> was again targeted, one day after a massive U.S. aerial bombardment in which more than 300 <i>Tomahawk</i> cruise <i>missiles</i> rained down on the <i>capital</i> .					
Event Trigger: targeted , Event Type: Attack					
Extraction Results:					
Entity	BIP	BIP(forward)	BIP(backward)	BIP-BI	BIP-SV
[Baghdad, capital]	Place(✓)	Place(✓)	Place(✓)	Place(✓)	Place(✓)
[Tomahawk, missiles]	None(✓)	Instrument(×)	None(✓)	None(✓)	None(✓)
Sentence 3: Last month, the <i>SEC</i> slapped fines totaling 1.4 billion dollars on 10 Wall Street <i>brokerages</i> to settle charges of conflicts of interest between analysts and investors.					
Event Trigger: fines , Event Type: Fine					
Extraction Results:					
Entity	BIP	BIP(forward)	BIP(backward)	BIP-BI	BIP-SV
SEC	Adjudicator(✓)	Adjudicator(✓)	Adjudicator(✓)	Adjudicator(✓)	Entity(×)
brokerages	Entity(✓)	Entity(✓)	Entity(✓)	Entity(✓)	Entity(✓)

Table 3: Event argument extraction results by different methods.

can observe that the variation of F1-score has a trend of rising with the increase of the training data. Comparing the fine-tuning method ONEIE, prompt-tuning methods BIP, BIP-BI and PAIE obviously improve the performance of role classification in low-resource data scenarios. This result shows that prompt-tuning methods can more effectively utilize the rich knowledge in PLMs than fine-tuning methods.

Even using flexible match to evaluate the prompt-tuning method PAIE, our method BIP and BIP-BI achieve better performance in both low-resource and high-resource data scenarios. The main reason is that our method can make use of the entity information and predicted argument roles when constructing the template. We notice that the performance of BIP is worse than that of BIP-BI, when the ratio of training data is less than 20%. This is because when the number of training data is too small, the probability of argument roles being correctly predicted is low. If the bi-directional iterative strategy is adopted, the wrongly predicted argument roles will be used for template construction, which will further degrade the performance of EAE.

4.7 Case Study

In order to showcase the effectiveness of our method BIP, we sample three sentences from the ACE 2005 English test dataset to compare the event argument extraction results by BIP, BIP(forward),

BIP(backward), BIP-BI and BIP-SV methods.

In Sentence 1 of Table 3, the method without the bi-directional iterative strategy BIP-BI can only identify the entity "**Bush**" as the "Entity" role. For the entity "**Putin**", the methods with the forward iterative prompt BIP, BIP(forward), BIP-SV can correctly classify it into the "Entity" role. This attributes to that the information that entity "**Bush**" is the "Entity" argument is introduced into the template construction of the entity "**Putin**". We also notice that "**Bush**" and "**Putin**" are both misclassified in the BIP(backward) method, where the error role information of "**Putin**" is passed to the classification of "**Bush**". In addition, for the entity "[**he, Erdogan**]" in Sentence 2, the method only with the forward iterative prompt BIP(forward) misclassifies the entity "[**Tomahawk, missiles**]" into the "Instrument" role. These results show that the argument roles of contextual entities can provide useful information for the role identification of the current entity. However, only considering argument interactions in one direction may degrade the performance of event argument extraction.

In Sentence 3, the method BIP-SV misclassifies the entity "**SEC**" into the "Entity" role. For the human-written verbalizer of BIP-SV, the word "judge" is selected as the label word of role "Adjudicator". It is difficult to associate the entity "**SEC**" with the word "judge". In the semantical verbalizer, we use the text sequence "the entity doing the fining" to describe the semantic of "Adjudicator"

Model	Template	Argument Identification			Role Classification		
		P	R	F1	P	R	F1
BIP(our)	Hard Template	78.17	86.40	82.08	75.26	83.19	79.03
	Soft Template	80.63	82.75	81.67	77.49	79.53	78.50
	Hard-Soft Template	77.15	82.46	79.72	74.15	79.24	76.61
BIP-BI	Hard Template	81.82	78.95	80.36	78.79	76.02	77.38
	Soft Template	76.25	84.94	80.36	73.62	82.02	77.59
	Hard-Soft Template	81.84	80.70	81.12	78.29	77.49	77.88

Table 4: Performance of different templates

role in the "Fine" event. Since the pre-trained language models can easily identify the entity "SEC" as "the entity doing the fining", the methods with semantical verbalizer can correctly identify the entity "SEC" as the "Adjudicator" role. The result verifies the effectiveness of our designed semantical verbalizer.

4.8 Prompt Variants

In this section, we compare three different templates introduced in Section 3.4 to investigate how different types of templates affect the performance of EAE. For the BIP-BI method, the performances of hard template, soft template and hard-soft template are comparable. Since the hard-soft template combines the manual knowledge and learnable virtual tokens, it achieves the best performance. However, the hard-soft template performs worst for the BIP method. Unlike the BIP-BI method which only considers event trigger and current entity information, BIP introduces the predicted argument role information into the template. Therefore, there are so many learnable pseudo tokens in the hard-soft template, resulting in poor performance.

5 Conclusion and Future Work

In this paper, we regard event argument extraction as a cloze-style task and propose a bi-directional iterative prompt-tuning method to address this task. The bi-directional iterative prompt-tuning method contains a forward iterative prompt and a backward iterative prompt, which predict the argument role of each entity in a left-to-right and right-to-left manner respectively. For the template construction in each prompt, the predicted argument role information is introduced to capture argument interactions. In addition, a novel semantical verbalizer is designed based on the semantic of the argument role. And three kinds of templates are designed and discussed. Experiment

results have shown the effectiveness of our method in both high-resource and low-resource data scenarios. In the future work, we are interested in the joint prompt-tuning method of event detection and event argument extraction.

Limitations

- As the entity information is necessary to model event argument extraction as a cloze-style task, our method is not suitable for the situation that entities are not provided.
- Comparing with the methods that predict argument roles simultaneously, the speed of our method is slower due to that it predicts the argument role of each entity one by one.

Acknowledgements

This work is supported in part by National Natural Science Foundation of China (Grant No: 62172167). We also want to use our BIP model on MindSpore², which is a new deep learning computing framework. These problems are left for future work.

References

- Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. [Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction](#). In *Proceedings of the ACM Web Conference 2022*, pages 2778–2788.
- 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.

²<http://www.mindspore.cn/>

- Beijing, China. Association for Computational Linguistics.
- Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. [Template-based named entity recognition using bart](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845.
- Lu Dai, Bang Wang, Wei Xiang, and Yijun Mo. 2021. [Event argument extraction via a distance-sensitive graph convolutional network](#). In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 59–72.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- George Doddington, Alexis Mitchell, Mark Przbocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. [The automatic content extraction \(ace\) program-tasks, data, and evaluation](#). In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 837–840.
- 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.
- Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. [Ptr: Prompt tuning with rules for text classification](#). *arXiv preprint arXiv:2105.11259*.
- Yutai Hou, Cheng Chen, Xianzhen Luo, Bohan Li, and Wanxiang Che. 2022. [Inverse is better! fast and accurate prompt for few-shot slot tagging](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 637–647, Dublin, Ireland. Association for Computational Linguistics.
- I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. [Degree: A data-efficient generative event extraction model](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.
- Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. [Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2225–2240, Dublin, Ireland. 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.
- 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.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. [Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing](#). *arXiv preprint arXiv:2107.13586*.
- Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022. [Dynamic prefix-tuning for generative template-based event extraction](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5216–5228, Dublin, Ireland. 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.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis,

- Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *arXiv preprint arXiv:1907.11692*.
- 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.
- Jie Ma, Shuai Wang, Rishita Anubhai, Miguel Ballesteros, and Yaser Al-Onaizan. 2020. [Resource-enhanced neural model for event argument extraction](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3554–3559, 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.
- 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.
- Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, RISHITA ANUBHAI, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2020. [Structured prediction as translation between augmented natural languages](#). In *International Conference on Learning Representations*.
- Timo Schick and Hinrich Schütze. 2021. [It’s not just size that matters: Small language models are also few-shot learners](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.
- Ronald Seoh, Ian Birle, Mrinal Tak, Haw-Shiuan Chang, Brian Pinette, and Alfred Hough. 2021. [Open aspect target sentiment classification with natural language prompts](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6311–6322, Online and Punta Cana, Dominican Republic. 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 32rd AAAI Conference on Artificial Intelligence*, pages 5916–5923.
- David Wadden, Ulme Wennberg, Yi Luan, and Hananeh Hajishirzi. 2019. [Entity, relation, and event extraction with contextualized span representations](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.
- Xiaozhi Wang, Ziqi Wang, Xu Han, Zhiyuan Liu, Juanzi Li, Peng Li, Maosong Sun, Jie Zhou, and Xiang Ren. 2019. [HMEAE: Hierarchical modular event argument extraction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5777–5783, Hong Kong, China. Association for Computational Linguistics.
- Xi Xiangyu, Wei Ye, Shikun Zhang, Quanxiu Wang, Huixing Jiang, and Wei Wu. 2021. [Capturing event argument interaction via a bi-directional entity-level recurrent decoder](#). 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 210–219, Online. Association for Computational Linguistics.

A Verbalizer

A.1 Semantical Verbalizer

For our designed semantical verbalizer, an argument role that participates in multiple types of events is divided into multiple argument roles that are specific to event types. For each new argument role, we use a virtual word to represent the role and initialize the representation of the virtual word with the semantic of the argument role. Table 6 shows the redefined argument role types, and the semantic description and virtual label word of each argument role type.

A.2 Human-written Verbalizer

For the human-written verbalizer, we assign a label word to each argument role. Table 6 lists the label word of each argument role.

B Templates

For our designed templates, each entity (event) type is converted into a human-understandable text span, so as to take full advantage of event type label and entity type label information. Table 7 and 8 list all text spans of entity types and event types.

Redefined Argument Role Label	Semantic Description	Virtual Label Word
Event:None	the entity that is irrelevant to the event	Event:None
Event:Place	the place where the event takes place	Event:Place
Be-Born:Person	the person who is born	Be-Born:Person
Marry:Person	the person who are married	Marry:Person
Divorce:Person	the person who are divorced	Divorce:Person
Injure:Agent	the one that enacts the harm	Injure:Agent
Injure:Victim	the harmed person	Injure:Victim
Injure:Instrument	the device used to inflict the harm	Injure:Instrument
Die:Agent	the killer	Die:Agent
Die:Victim	the person who died	Die:Victim
Die:Instrument	the device used to kill	Die:Instrument
Transport:Agent	the agent responsible for the transport event	Transport:Agent
Transport:Artifact	the person doing the traveling or the artifact being transported	Transport:Artifact
Transport:Vehicle	the vehicle used to transport the person or artifact	Transport:Vehicle
Transport:Origin	the place where the transporting originated	Transport:Origin
Transport:Destination	the place where the transporting is directed	Transport:Destination
Transfer-Ownership:Buyer	the buying agent	Transfer-Ownership:Buyer
Transfer-Ownership:Seller	the selling agent	Transfer-Ownership:Seller
Transfer-Ownership:Beneficiary	the agent that benefits from the transaction	Transfer-Ownership:Beneficiary
Transfer-Ownership:Artifact	the item or organization that was bought or sold	Transfer-Ownership:Artifact
Transfer-Money:Giver	the donating agent	Transfer-Money:Giver
Transfer-Money:Recipient	the recipient agent	Transfer-Money:Recipient
Transfer-Money:Beneficiary	the agent that benefits from the transfer	Transfer-Money:Beneficiary
Start-Org:Agent	the founder	Start-Org:Agent
Start-Org:Org	the organization that is started	Start-Org:Org
Merge-Org:Org	the organizations that are merged	Merge-Org:Org
Declare-Bankruptcy:Org	the organization declaring bankruptcy	Declare-Bankruptcy:Org
End-Org:Org	the organization that is ended	End-Org:Org
Attack:Attacker	the attacking agent	Attack:Attacker
Attack:Target	the target of the attack	Attack:Target
Attack:Instrument	the instrument used in the attack	Attack:Instrument
Demonstrate:Entity	the demonstrating agent	Demonstrate:Entity
Meet:Entity	the agents who are meeting	Meet:Entity
Phone-Write:Entity	the communicating agent	Phone-Write:Entity
Start-Position:Person	the employee	Start-Position:Person
Start-Position:Entity	the employer	Start-Position:Entity
End-Position:Person	the employee	End-Position:Person
End-Position:Entity	the employer	End-Position:Entity
Elect:Person	the person elected	Elect:Person
Elect:Entity	the voting agent	Elect:Entity
Nominate:Person	the person nominated	Nominate:Person
Nominate:Agent	the nominating agent	Nominate:Agent
Arrest-Jail:Person	the person who is jailed or arrested	Arrest-Jail:Person
Arrest-Jail:Agent	the jailer or the arresting agent	Arrest-Jail:Agent
Release-Parole:Person	the person who is released	Release-Parole:Person
Release-Parole:Entity	the former captor agent	Release-Parole:Entity
Trial-Hearing:Defendant	the agent on trial	Trial-Hearing:Defendant
Trial-Hearing:Prosecutor	the prosecuting agent	Trial-Hearing:Prosecutor
Trial-Hearing:Adjudicator	the judge or court	Trial-Hearing:Adjudicator
Charge-Indict:Defendant	the agent that is indicted	Charge-Indict:Defendant
Charge-Indict:Prosecutor	the agent bringing charges or executing the indictment	Charge-Indict:Prosecutor
Sue:Plaintiff	the suing agent	Sue:Plaintiff
Sue:Defendant	the agent being sued	Sue:Defendant
Sue:Adjudicator	the judge or court	Sue:Adjudicator
Convict:Defendant	the convicted agent	Convict:Defendant
Convict:Adjudicator	the judge or court	Convict:Adjudicator
Sentence:Defendant	the agent who is sentenced	Sentence:Defendant
Sentence:Adjudicator	the judge or court	Sentence:Adjudicator
Fine:Entity	the entity that was fined	Fine:Entity
Fine:Adjudicator	the entity doing the fining	Fine:Adjudicator
Execute:Person	the person executed	Execute:Person
Execute:Agent	the agent responsible for carrying out the execution	Execute:Agent
Extradite:Person	the person being extradited	Extradite:Person
Extradite:Agent	the extraditing agent	Extradite:Agent
Extradite:Origin	the original location of the person being extradited	Extradite:Origin
Extradite:Destination	the place where the person is extradited to	Extradite:Destination
Acquit:Defendant	the agent being acquitted	Acquit:Defendant
Acquit:Adjudicator	the judge or court	Acquit:Adjudicator
Pardon:Defendant	the agent being pardoned	Pardon:Defendant
Pardon:Adjudicator	the state official who does the pardoning	Pardon:Adjudicator
Appeal:Defendant	the defendant	Appeal:Defendant
Appeal:Adjudicator	the judge or court	Appeal:Adjudicator
Appeal:Plaintiff	the appealing agent	Appeal:Plaintiff

Table 5: Label words of the human-written verbalizer.

Argument Role Label	Label Word
None	none
Person	person
Place	place
Buyer	buyer
Seller	seller
Beneficiary	beneficiary
Artifact	artifact
Origin	origin
Destination	destination
Giver	donor
Recipient	recipient
Org	organization
Agent	agent
Victim	victim
Instrument	instrument
Entity	entity
Attacker	attacker
Target	target
Defendant	defendant
Adjudicator	judge
Prosecutor	prosecutor
Plaintiff	plaintiff
Vehicle	vehicle

Table 6: Label words of the human-written verbalizer.

Event Type	Text Span
Transport	transport
Elect	election
Start-Position	employment
End-Position	dimission
Attack	attack
Meet	meeting
Marry	marriage
Transfer-Money	money transfer
Demonstrate	demonstration
End-Org	collapse
Sue	prosecution
Injure	injury
Die	death
Arrest-Jail	arrest or jail
Phone-Write	written or telephone communication
Transfer-Ownership	ownership transfer
Start-Org	organization founding
Execute	execution
Trial-Hearing	trial or hearing
Be-Born	birth
Charge-Indict	charge or indict
Sentence	sentence
Declare-Bankruptcy	bankruptcy
Release-Parole	release or parole
Fine	fine
Pardon	pardon
Appeal	appeal
Extradite	extradition
Divorce	divorce
Merge-Org	organization merger
Acquit	acquittal
Nominate	nomination
Convict	conviction

Table 8: Text spans of event types.

Entity Type	Text Span
FAC	facility
ORG	organization
GPE	geographical or political entity
PER	person
VEH	vehicle
WEA	weapon
LOC	location

Table 7: Text spans of entity types.