Zero-Shot Cross-Lingual Event Argument Extraction with Language-Oriented Prefix-Tuning

Pengfei Cao[1,2*], **Zhuoran Jin**[1,2*], **Yubo Chen**[1,2], **Kang Liu**[1,2,3], **Jun Zhao**[1,2]

[1]National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3]Beijing Academy of Artificial Intelligence, Beijing, 100084, China
{pengfei.cao, zhuoran.jin, yubo.chen, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

Event argument extraction (EAE) aims to identify the arguments of a given event, and classify the roles that those arguments play. Due to high data demands of training EAE models, zero-shot cross-lingual EAE has attracted increasing attention, as it greatly reduces human annotation effort. Some prior works indicate that generation-based methods have achieved promising performance for monolingual EAE. However, when applying existing generation-based methods to zero-shot cross-lingual EAE, we find two critical challenges, including *Language Discrepancy* and *Template Construction*. In this paper, we propose a novel method termed as **L**angu**A**ge-oriented **P**refix-tun**I**ng **N**etwork (**LAPIN**) to address the above challenges. Specifically, we devise a *Language-oriented Prefix Generator* module to handle the discrepancies between source and target languages. Moreover, we leverage a *Language-agnostic Template Constructor* module to design templates that can be adapted to any language. Extensive experiments demonstrate that our proposed method achieves the best performance, outperforming the previous state-of-the-art model by 4.8% and 2.3% of the average F1-score on two multilingual EAE datasets.

Introduction

Event argument extraction (EAE) aims to identify the entities that serve as event arguments of a given event, and predict the roles they play. Figure 1(a) illustrates an example of EAE task. Given the trigger "*attacked*" for an *Attack* event, the EAE model is expected to recognize "*two soldiers*", "*demonstrators*" and "*yesterday*" as the event arguments, and predict their roles as "*Target*", "*Attacker*" and "*Time*", respectively. EAE is a key step for event extraction (EE), which can be beneficial for various downstream applications, such as recommendation systems (Li et al. 2020b), dialogue systems (Zhang, Chen, and Bui 2020) and timeline summarization (Li et al. 2021).

With the development of deep neural network methods, EAE has met with remarkable success. However, training an EAE model relies on considerably large-scale labeled data, which makes it hard to adapt to low-resource languages that lack sufficient labeled data. To address this shortcoming, cross-lingual EAE transfers the advantage of the source language with rich resources to the target language, which gains increasing attention (Subburathinam et al. 2019; Van Nguyen and Nguyen 2021; Ahmad, Peng, and Chang 2021; Lou et al. 2022). In this work, we focus on the zero-shot cross-lingual EAE task, where the EAE model is trained with the annotated data in a source language and directly applied to other target languages (*cf.* Figure 1(c)).

For monolingual EAE, most of previous methods treat it as classification tasks (including entity recognition and argument classification), either trained in a pipelined or joint manner (Nguyen, Cho, and Grishman 2016; Wang et al. 2019; Lin et al. 2020). Recently, there is an emerging trend of casting EAE as a sequence generation problem (Li, Ji, and Han 2021; Hsu et al. 2022; Liu et al. 2022). As shown in Figure 1(b), this line of methods design templates with several placeholders (i.e., underlined words), and steer generative pre-trained language models (PLMs) to fill them. Com-

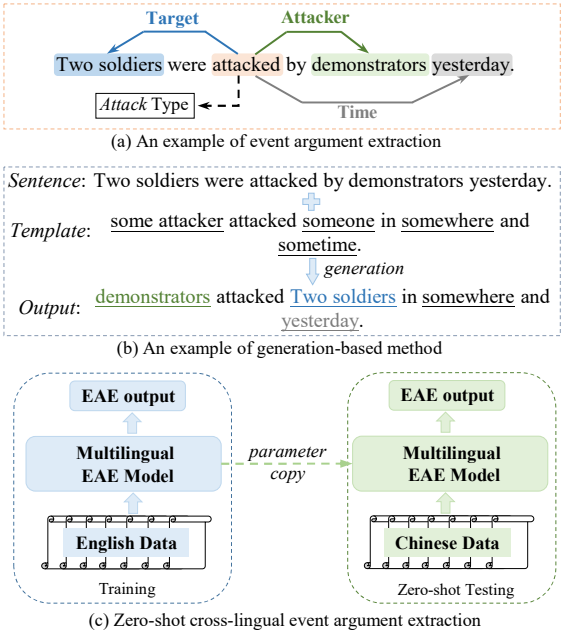

Figure 1: (a) An example of event argument extraction; (b) An example of generation-based method; (c) An illustration of zero-shot cross-lingual event argument extraction.

*These authors contributed equally.

pared with classification-based methods, generation-based methods can better capture dependencies between arguments, which achieves promising performance for monolingual EAE. Despite the success, extending generation-based methods to the zero-shot cross-lingual EAE is non-trivial, which faces two critical challenges: (1) **Language Discrepancy**. Languages have their own characteristics, e.g., the distance between triggers and candidate arguments is very different among languages. If the EAE model trained in source languages memorizes excessive specific knowledge from source languages, it inevitably has a negative impact on the prediction in target languages. For example, according to our statistics, the average distance between triggers and candidate arguments in English is 9.8, while it is 21.7 in Chinese. As a result, an EAE model trained in English may pay too much attention on closer tokens, thus failing to generalize in Chinese. (2) **Template Construction**. Designing templates is a very important step in generation-based methods. As shown in Figure 1(b), the templates designed in prior works are language-dependent, i.e., the language of templates is the same as that of training instances. However, for zero-shot cross-lingual EAE, the languages of instances are different during training and testing. Thus, the language-dependent templates severely restrict the zero-shot cross-lingual transfer of existing generation-based methods. Naively applying such models trained in source languages to target languages usually generates words belonging to source languages, yielding poor performance.

In this paper, we propose a novel method termed as **L**angu**A**ge-oriented **P**refix-tun**I**ng **N**etwork (**LAPIN**) to address the aforementioned challenges. Our method is based on the multilingual generative PLMs for conditional generation. Specifically, we devise a *Language-oriented Prefix Generator* module to handle the discrepancies between source and target languages. The module first obtains and encodes the language-universal dependency structure of the input sentence, and then utilizes trigger-centric neighbor information to initialize continuous prefix vectors. Meanwhile, inspired by Huang et al. (2022), we use a *Language-agnostic Template Constructor* module to facilitate cross-lingual transfer. The module utilizes some language-agnostic tokens to represent templates, so that the constructed templates can be adapted to any language. Extensive experiments on two multilingual EAE datasets demonstrate that our method substantially outperforms previous state-of-the-art zero-shot cross-lingual EAE models.

Overall, the contributions of this work can be summarized as follows:

- We propose a novel language-oriented prefix-tuning network (LAPIN) for zero-shot cross-lingual EAE. To our best knowledge, we are the first to explore the prefix-tuning method for the task.
- We introduce a language-oriented prefix initialization mechanism based on the language-universal dependency structure, which can help the model handle the discrepancies between source and target languages.
- Experimental results indicate that our approach significantly outperforms previous state-of-the-art methods, achieving 4.8% and 2.3% improvements of average F1-score on two widely used datasets.

Related Work

Traditional Event Argument Extraction

Event argument extraction (EAE) is an important subtask of event extraction (EE), which has attracted extensive attention among researchers. Existing methods for EAE can be mainly classified into two categories. The first category of methods formulates EAE as a classification problem (Chen et al. 2015; Huang et al. 2018). These methods usually first identify candidate arguments and then predict their roles. In addition, recent works apply reading comprehension methods to the EAE task (Du and Cardie 2020; Li et al. 2020a; Liu et al. 2020). In the second category, recent studies treat EAE as a sequence generation problem (Li, Ji, and Han 2021; Hsu et al. 2022; Liu et al. 2022), with the help of generative PLMs (Lewis et al. 2020; Raffel et al. 2020). Compared with classification-based methods, generation-based methods have achieved more promising results by capturing dependencies between triggers and arguments.

Cross-Lingual Event Argument Extraction

The success of traditional EAE is almost limited to high-resource languages, which requires an amount of annotated data for training. To alleviate this problem, zero-shot cross-lingual EAE has gained increasing attention in recent years (Subburathinam et al. 2019; Huang et al. 2022). Most previous works on cross-lingual transfer for EE are based on machine translation (Zhu et al. 2014) and external resources or data (Chen and Ji 2009; Hsi et al. 2016). Subburathinam et al. (2019) leverage graph neural networks (Kipf and Welling 2017) to learn multilingual representations across languages. In view of the success of generation-based methods for monolingual EAE, Huang et al. (2022) extend the idea to the cross-lingual setting, which achieves the current best performance for zero-shot cross-lingual EAE. However, the method cannot effectively handle discrepancies between source and target languages, and easily overfits the specific information of source languages.

Prompt-based Learning

With the success of PLMs, prompt-based learning has become a promising paradigm (Liu et al. 2021). It transforms the downstream tasks into the same form as the PLMs' pre-training tasks, which is more helpful to elicit model knowledge (Brown et al. 2020). For example, Han et al. (2022) convert text classification problems to cloze-style tasks, which depends on designed verbalizers to map from label words to specific labels. These discrete prompt-tuning methods are effective for few-shot classification tasks (Schick and Schütze 2021; Seoh et al. 2021). Additionally, instead of discrete prompts, some studies also propose continuous prompts that are directly operated in the embedding space (Li and Liang 2021; Lester, Al-Rfou, and Constant 2021; Qin and Eisner 2021). Despite the flourish of the research in prompt-based learning, how to apply it to the zero-shot cross-lingual EAE remains largely under-explored.

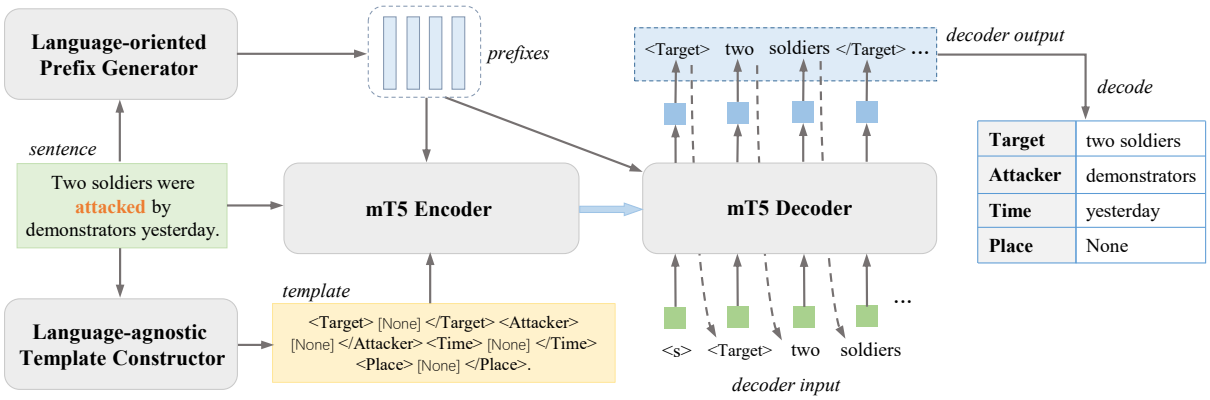

Figure 2: The architecture of our proposed language-oriented prefix-tuning network (LAPIN) for zero-shot cross-lingual EAE.

Task Formulation

Following Hsu et al. (2022) and Huang et al. (2022), we formulate event argument extraction task as follows. We define that an input sentence $\mathcal{S}$ consists of T words, i.e., $\mathcal{S} = \{w_1, w_2, \ldots, w_T\}$. We also define the event types set $\mathcal{E}$ and the corresponding argument roles set $\mathcal{R}_e$ for each event type $e \in \mathcal{E}$. Given an input sentence $\mathcal{S}$ and an event trigger t belonging to an event type $e \in \mathcal{E}$, the EAE task aims to recognize all $(r, s) \in \mathcal{A}$ pairs for the event, where $r \in \mathcal{R}_e$ is an argument role for the event type e, and $s \in \mathcal{S}$ is a contiguous text span in the sentence. For the zero-shot cross-lingual EAE, the training data $\mathcal{D}_{train} = \{(\mathcal{S}_i, t_i, e_i, \mathcal{R}_{e_i}, \mathcal{A}_i)\}_{i=1}^{N}$ belongs to the source languages, which is used to train an EAE model. Then, the trained model is directly tested on instances of target languages, denoted as $\mathcal{D}_{test} = \{(\mathcal{S}_i, t_i, e_i, \mathcal{R}_{e_i}, \mathcal{A}_i)\}_{i=1}^{M}$.

In contrast to monolingual EAE, zero-shot cross-lingual EAE is a more challenging yet practical problem. It requires EAE models to be capable of transferring the shared knowledge from the source languages to the target languages.

Methodology

Figure 2 shows the overall architecture of LAPIN, which consists of three major components: (1) *Language-agnostic Template Constructor*, which designs templates according to event structures; (2) *Encoder-Decoder Architecture*, which leverages multilingual generative PLMs to fill the template; (3) *Language-oriented Prefix Generator*, which initializes prefixes using language-universal dependency structure. We will illustrate each component in detail.

Language-agnostic Template Constructor

Previous works (Hsu et al. 2022; Ma et al. 2022) have proved that templates are very important for generation-based methods. Generally, for each event type e, a type-specific template $\mathcal{T}_e$ should be designed according to event structures (i.e., ontologies). It usually contains several placeholders that need to be replaced by concrete arguments. In addition, the language of instances may be different during training and testing for zero-shot cross-lingual EAE. Therefore, the template should be designed in a language-agnostic manner.

Following Huang et al. (2022), we utilize a unique HTML-tag-style template, which can meet the above two requirements. For example, the *Attack* event is associated with four roles, including *Target*, *Attacker*, *Time* and *Place*. The template for *Attack* events is designed as:

⟨Target⟩ [None] ⟨/Target⟩ ⟨Attacker⟩ [None] ⟨/Attacker⟩ ⟨Time⟩ [None] ⟨/Time⟩ ⟨Place⟩ [None] ⟨/Place⟩.

In the template, the special token [None] serves as the argument placeholder. Other special tokens (e.g., ⟨Target⟩, ⟨/Target⟩) are unseen for PLMs during pre-training stage. Their representations can be learned from scratch using the training data, which naturally captures the information of event structures. Since these special tokens do not belong to any language, the constructed template can be considered language-agnostic. In this way, the templates of other event types can also be constructed.

To construct the ground truth output sequence, we replace the placeholder [None] in the template with the corresponding gold arguments. If there are no corresponding arguments for one role in the input sentence, we keep [None] in the template. For example in Figure 1, the target output sequence of the example is:

⟨Target⟩ two soldiers ⟨/Target⟩ ⟨Attacker⟩ demonstrators ⟨/Attacker⟩ ⟨Time⟩ yesterday ⟨/Time⟩ ⟨Place⟩ [None] ⟨/Place⟩.

In addition, if more than one argument is predicted as the same role, they are first sorted by spans and then connected by the special token [and]. Given the generated output sequence, we can easily parse the argument and role predictions according to event structures.

Encoder-Decoder Architecture

Given the input sentence $\mathcal{S}$, the template $\mathcal{T}_e$ is designed using the above template construction strategy. Our method LAPIN generates the output sequence $\mathcal{Y}$ via multilingual generative PLMs (i.e., mT5 (Xue et al. 2021)). Concretely, our method first encodes the input sequence and obtains corresponding representations:

$$\boldsymbol{H}_{\mathcal{X}} = \text{Encoder}(\mathcal{X}), \quad \mathcal{X} = [\mathcal{S}; [\text{SEP}]; \mathcal{T}_e], \qquad (1)$$

where Encoder$(\cdot)$ is a multi-layer transformer encoder (Vaswani et al. 2017). $\mathcal{X}$ denotes the input sequence that is concatenated by the sentence $\mathcal{S}$ and template $\mathcal{T}_e$. [SEP] denotes the separate marker in the PLMs. $[;]$ indicates the sequence concatenation operation. $\boldsymbol{H}_{\mathcal{X}} \in \mathbb{R}^{|\mathcal{X}| \times d}$ denote the hidden representations for each token in the input sequence.

After obtaining hidden representations of the input sequence, we feed them into the decoder for generating the output sequence $\mathcal{Y}$ in an autoregressive style (i.e., token-by-token). At step t, the decoder generates the t-th token y_t and decoder state $\boldsymbol{h}_t^d$ as follows:

$$y_t, \boldsymbol{h}_t^d = \text{Decoder}(\boldsymbol{H}_{\mathcal{X}}, \boldsymbol{H}_{<t}^d, y_{t-1}), \tag{2}$$

where Decoder$(\cdot)$ is a multi-layer transformer decoder. $\boldsymbol{H}_{<t}^d \in \mathbb{R}^{(t-1) \times d}$ are past states of the decoder during decoding. The conditional probability of the entire output sequence, denoted as $p(\mathcal{Y} \mid \mathcal{X})$, can be computed as follows:

$$p(\mathcal{Y}|\mathcal{X}) = \prod_{t=1}^{|\mathcal{Y}|} p(y_t \mid y_{<t}, \mathcal{X}), \tag{3}$$

where $p(y_t \mid y_{<t}, \mathcal{X})$ is the probability of predicting token y_t, given the previous generated tokens $y_{<t}$ and the encoder input $\mathcal{X}$.

Language-oriented Prefix Generator

To alleviate the language discrepancies, we devise a language-oriented prefix generator to initialize prefixes based on a language-universal dependency structure, which is illustrated in Figure 3. It guides the model to learn shared knowledge between source and target languages. Specifically, it is designed as follows:

Encoding of Dependency Structure We first use a pre-trained universal dependency parser (e.g., Stanza[1] (Qi et al. 2020)) to obtain the language-universal dependency tree of the input sentence. Then, we compute the syntactic (i.e., shortest path on the tree) distance between every pair of tokens. The distance matrix is denoted as $\boldsymbol{D} \in \mathbb{R}^{|T| \times |T|}$, where $\boldsymbol{D}_{ij}$ represents the syntactic distance between i-th and j-th tokens in the input sentence, and T denotes the length of the input sentence.

To encode the dependency structure, we first use the mT5 encoder to represent each token in a shared semantic space across languages, and then employ the transformer as the structure encoder. If tokens is allowed to attend other tokens that are within distance δ, the mask can be defined matrix as follows:

$$\boldsymbol{M}_{ij} = \begin{cases} 0, & \text{if } \boldsymbol{D}_{ij} \leq \delta \\ -\infty, & \text{otherwise,} \end{cases} \tag{4}$$

where δ is a hyper-parameter. In this way, the mask matrix can take into account the syntactic structure. For l-th transformer layer, the self-attention distribution $\boldsymbol{P}_l$ is computed as follows:

$$\boldsymbol{P}^l = \text{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}} + \boldsymbol{M})\boldsymbol{V}, \tag{5}$$

[1]https://stanfordnlp.github.io/stanza/

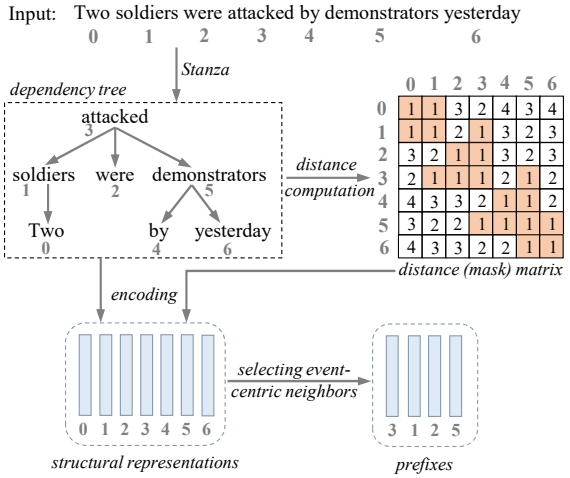

Figure 3: An illustration of language-oriented prefix generator. In the example, the distance δ is 1 and the length of the prefix is 4. For the distance (mask) matrix, orange means visible (i.e., $\boldsymbol{M}_{ij}$=0), and white means invisible.

where $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ are queries, keys and values of the l-th layer, respectively, whose hidden size is d_k. $\boldsymbol{P}_{ij}^l$ denotes the attention that i-th token pays to the j-th token in the sentence. Since the syntactic distances between trigger and arguments are informative (Ahmad, Peng, and Chang 2021), we modify the self-attention distribution by incorporating syntactic distances:

$$\boldsymbol{A}_{ij}^l = \frac{\boldsymbol{P}_{ij}^l}{\boldsymbol{Z}_i \boldsymbol{D}_{ij}}, \tag{6}$$

where $\boldsymbol{Z}_i = \sum_j \frac{\boldsymbol{P}_{ij}}{\boldsymbol{D}_{ij}}$ is the normalization factor. $\boldsymbol{A}^l \in \mathbb{R}^{|T| \times |T|}$ is the revised self-attention matrix, which is used to compute the l-th layer output of the transformer.

Selecting Trigger-centric Neighbors After the encoding, we obtain the structural representations of each token. The dependency tree is transformed into an undirected graph. We take triggers as the starting point, and select a certain number of neighbors in the order of breadth-first search, as shown in Figure 3. The selection strategy can not only indicate the trigger position, but also capture the dependencies between triggers and arguments. These selected token representations compose a learnable matrix $\boldsymbol{U} \in \mathbb{R}^{L \times d}$, where L is the number of selected tokens. It serves as initialized representations of prefixes, which are prepended for the sequences $\mathcal{X}$ and $\mathcal{Y}$ of each transformer layer in encoder and decoder. With the injection of prefixes $\boldsymbol{U}$, the computation of decoder state $\boldsymbol{h}_t^d$ in Equation (2) is modified as follows[2]:

$$\boldsymbol{h}_t^d = \begin{cases} \boldsymbol{U}[t,:], & \text{if } t < L \\ \text{Decoder}(\boldsymbol{H}_{\mathcal{X}}, \boldsymbol{H}_{<t}^d, y_{t-1}), & \text{otherwise.} \end{cases} \tag{7}$$

The computation of encoder states is similar. In this way, we induce the model to capture language-shared information.

[2]For simplicity, we only present the decoder state $\boldsymbol{h}_t^d$, not the t-th generated token y_t in the equation.

Models	PLMs	en ⇓ en	en ⇓ zh	en ⇓ ar	ar ⇓ ar	ar ⇓ en	ar ⇓ zh	zh ⇓ zh	zh ⇓ en	zh ⇓ ar	avg
Classification-based Methods											
OneIE	XLM-R-large	63.6	42.5	37.5	57.8	27.5	31.2	69.6	51.5	31.1	45.8
CL-GCN	XLM-R-large	59.8	29.4	25.0	47.5	25.4	19.4	62.2	40.8	23.3	37.0
GATE	XLM-R-large	67.0	49.2	44.5	59.6	27.6	26.3	70.6	46.7	37.3	47.6
GATE	mBART-50-large	65.5	43.0	38.9	58.5	27.5	26.1	65.9	45.3	30.2	44.5
GATE	mT5-base	59.8	47.7	32.6	45.4	20.7	21.0	64.0	35.3	22.8	38.8
Generation-based Methods											
TANL	mT5-base	59.1	38.6	29.7	50.1	18.3	16.9	65.2	33.3	18.3	36.6
X-GEAR	mT5-base	67.9	53.1	42.0	66.2	27.6	30.5	69.4	52.8	32.0	49.1
X-GEAR	mT5-large	71.2	54.0	44.8	68.9	32.1	33.3	68.9	55.8	33.1	51.3
Our Proposed Method											
LAPIN	mT5-base	69.0	57.1	41.8	67.0	29.5	36.0	68.0	55.3	36.2	51.1
LAPIN	mT5-large	**74.4**	**59.3**	**52.0**	**69.4**	**36.8**	**44.3**	**72.5**	**59.1**	**37.4**	**56.1**

Table 1: Experimental results (F1-score, %) of different models on the ACE-2005 dataset. The languages on top and bottom of ⇓ denote the source language and target language, respectively. "avg" denotes the average performance of all the combinations of the source language and the target language. Bold denotes best results.

Training

The trainable parameters of our method contain the parameters of the encoder-decoder model and generated prefixes, which is denoted as θ. We use the negative log-likelihood function to optimize the model:

$$\mathcal{L}_\theta(\mathcal{D}) = -\sum_{i=1}^{|\mathcal{D}|} \log p(\mathcal{G}_i \mid \mathcal{X}_i, \theta), \tag{8}$$

where $\mathcal{D}$ denotes the training set in the source languages. $\mathcal{X}_i$ is the input sequence of i-th example, and $\mathcal{G}_i$ is the corresponding ground truth output sequence.

Given that most of the tokens in the target output sequence are also present in the input sequence, we augment the multilingual generative PLMs with a copy mechanism (See, Liu, and Manning 2017), which can help our method LAPIN better adapt to the cross-lingual scenario.

Experiments

Datasets and Evaluation Metrics

We evaluate our method on two EE datasets, including ACE-2005 (Doddington et al. 2004) and ERE (Song et al. 2015). For ACE-2005, the dataset is labeled in three languages: English (en), Chinese (zh) and Arabic (ar). For a fair comparison with previous work (Huang et al. 2022), we use the same dataset split and prepossessing methods to keep 33 event types and 22 argument roles. For ERE, the dataset is annotated in two languages: English and Spanish (es). Following the preprocessing in Lin et al. (2020) and Huang et al. (2022), we keep 38 event types and 21 argument roles.

Following previous works (Ahmad, Peng, and Chang 2021; Huang et al. 2022), we use the F1-score of argument classification as the evaluation metric. If argument offsets and role type are both same as the ground truth, an argument-role pair is assumed to be correctly classified. For the offset of the predicted argument, we select the nearest matched string to the predicted trigger, same as previous methods (Huang et al. 2022) to ensure fairness.

Parameter Settings

In our implementations, our method uses the HuggingFace's Transformers library[3] to implement the encoder-decoder mT5 (base and large) model. To embed tokens of the dependency structure into vector representations, we use another mT5 encoder as a feature extractor and do not fine-tune it. The learning rate is initialized as 3e-5 or 1e-4 with a linear decay for mT5-base or mT5-large models, respectively. We utilize the AdamW algorithm (Loshchilov and Hutter 2017) to optimize model parameters. The batch size is set to 8. Our method generates output sequences by using beam search, whose beam size is set to 4. The length of prefix L is set to 30. The distance hyper-parameter δ is set to 2. The number of training epochs is 100. Each experiment is conducted on NVIDIA RTX A6000 GPUs.

Baselines

We compare the proposed approach LAPIN with the following methods:

(1) **OneIE** (Lin et al. 2020), which is a classification-based monolingual information extraction model. It achieves very competitive performance for the EAE task. Following Huang et al. (2022), we employ the XLM-RoBERTa-large (XLM-R-large) (Conneau et al. 2020) to obtain the word embedding for each token, so that the model can adapt to the zero-shot cross-lingual setting.

(2) **CL-GCN** (Subburathinam et al. 2019), which is proposed to address event argument role labeling (EARL). It belongs to classification-based models and uses graph convolutional networks to encode the dependency structure. Since EARL requires that the entities are given in advance, one named entity recognition module is required for the model.

[3]https://github.com/huggingface/transformers

Models	en ⇓ en	ar ⇓ ar	zh ⇓ zh	avg_m	en ⇓ ar	en ⇓ zh	ar ⇓ en	ar ⇓ zh	zh ⇓ ar	zh ⇓ en	avg_c	avg_a
LAPIN (mT5-base)	69.0	67.0	**68.0**	**68.0**	**41.8**	**57.1**	**29.5**	**36.0**	36.2	**55.3**	**42.7**	**51.1**
w/o prefix-tuning	68.1	64.8	67.5	66.8 (↓1.2)	40.3	55.8	23.2	33.1	33.7	53.4	39.9 (↓2.8)	48.8 (↓2.3)
w/ sequential selection	**69.1**	**67.1**	64.8	67.0 (↓1.0)	40.1	56.0	28.2	35.6	**36.7**	52.3	41.5 (↓1.2)	50.0 (↓1.1)
LAPIN (mT5-large)	**74.4**	69.4	72.5	**72.1**	**52.0**	59.3	**36.8**	**44.3**	**37.4**	**59.1**	**48.2**	**56.1**
w/o prefix-tuning	71.6	**69.9**	**72.8**	71.4 (↓0.7)	49.1	59.0	35.4	40.7	36.0	56.1	46.1 (↓2.1)	54.5 (↓1.6)
w/ sequential selection	72.1	68.3	71.4	70.6 (↓1.5)	49.5	**62.6**	32.4	43.0	37.3	58.0	47.1 (↓1.1)	55.0 (↓1.1)

Table 2: Ablation study of language-oriented prefix-tuning on the ACE-2005 dataset. "avg_m" indicates the average of monolingual settings (i.e., "en ⇒ en", "ar ⇒ ar", and "zh ⇒ zh"). "avg_c" indicates the average of cross-lingual settings. "avg_a" indicates the average of all the combinations of the source language and the target language. The average F1-score is followed by the drop (↓) compared with the method LAPIN.

Models	PLMs	en ⇓ en	en ⇓ es	es ⇓ es	es ⇓ en	avg
Classification-based Methods						
OneIE	XLM-R-large	64.4	56.8	64.8	56.9	60.7
CL-GCN	XLM-R-large	61.9	51.9	62.9	48.5	55.9
GATE	XLM-R-large	66.4	61.5	63.0	56.5	61.9
Generation-based Methods						
TANL	mT5-base	65.9	40.3	58.6	47.4	53.1
X-GEAR	mT5-base	69.8	57.9	66.1	59.0	63.2
X-GEAR	mT5-large	72.9	59.7	67.4	64.1	66.0
Our Proposed Method						
LAPIN	mT5-base	71.6	59.8	67.5	61.4	65.1
LAPIN	mT5-large	**73.1**	**64.6**	**69.6**	**66.0**	**68.3**

Table 3: Experimental results (F1-score, %) of different models on the ERE dataset.

(3) **GATE** (Ahmad, Peng, and Chang 2021), which is a classification-based zero-shot cross-lingual EARL model. Unlike CL-GCN, the model leverage the transformer to encode the dependency structure. Following Huang et al. (2022), we utilize the multilingual PLMs to obtain the representation of each token.

(4) **TANL** (Paolini et al. 2021), which is a generation-based EAE model. The model inserts the role labels into the input sentence to obtain the target sequence. It is originally proposed for monolingual EAE task based on the T5. To accommodate zero-shot cross-lingual EAE for the TANL model, we replace the T5 with mT5-base.

(5) **X-GEAR** (Huang et al. 2022), which is a generation-based zero-shot cross-lingual EAE model. The model is based on multilingual PLMs, including mT5-base and mT5-large. It aims to devise a language-universal template, which achieves the current best performance for the zero-shot cross-lingual EAE task.

Overall Results

Table 1 and Table 3 shows the results on the ACE-2005 and ERE datasets, respectively. We note the following key observations throughout our experiments:

(1) Our method outperforms all the baselines by a large margin, and achieves new state-of-the-art performance on the two datasets. For example, compared with the previous state-of-the-art model X-GEAR (mT5-large), our method achieves 4.8% and 2.3% improvements of average F1-score on the ACE-2005 and ERE datasets, respectively. The significant performance gain of our method over the baselines demonstrates that the proposed method LAPIN is very effective for the zero-shot cross-lingual EAE task.

(2) Compared with classification-based methods, our approach achieves greater improvements. For example, our method outperforms the classification-based model GATE (XLM-R-large) by 8.5% and 6.4% in term of average F1-score on the ACE-2005 and ERE datasets, respectively. We attribute the improvements to that our method LAPIN takes advantage of argument dependencies and language-universal knowledge, thus achieving superior performance.

(3) The generation-based model TANL yields worse performance than our method LAPIN and X-GEAR. The reason is that the language-dependent template is not suitable for the cross-lingual setting. In addition, our method with mT5-large achieves better performance than that with mT5-base. It suggests that the performance of LAPIN can be further improved with larger generative PLMs.

Effectiveness of Language-oriented Prefix-tuning

To demonstrate the effectiveness of the language-oriented prefix-tuning, we conduct an ablation study as follows. 1) w/o prefix-tuning, which removes the language-oriented prefix generator module from our method; 2) w/ sequential selection, which selects tokens as prefixes in sequential order starting with the first token of the sentence, instead of the breadth-first search order. We present the results of ablation study in Table 2. From the results, we can observe that:

(1) When we remove the language-oriented prefix-tuning, the performance drops significantly in all the scenarios. The average all of the F1-score (i.e., avg_a) drops by 2.3% over the LAPIN (mT5-base). It is worth noting that the decline of cross-lingual settings (e.g., ↓2.8 of avg_c) is greater than that of monolingual settings (e.g., ↓1.2 of avg_m). It indicates that the language-oriented prefix-tuning is able to handle the discrepancies between different languages.

Models	(0,5]	(5,10]	(10,15]	(15,20]	(20,30]	(30,40]
			en ⇒ *xx*			
X-GEAR	59.3	47.8	40.6	29.6	53.7	**35.2**
LAPIN	**62.9**	**56.5**	**52.5**	**53.7**	**60.8**	28.2
			ar ⇒ *xx*			
X-GEAR	43.7	36.1	25.5	32.1	**11.5**	9.5
LAPIN	**47.2**	**43.7**	**41.7**	**41.4**	10.6	**26.7**
			zh ⇒ *xx*			
X-GEAR	57.0	47.8	46.9	41.7	22.3	28.6
LAPIN	**57.2**	**52.9**	**53.5**	**53.6**	**60.5**	**37.8**

Table 4: F1-score on the ACE-2005 dataset with different distance spans between triggers and arguments. The instance of distance over 40 is very few, thus we ignore the case. "en ⇒ xx" indicates the performance average of "en ⇒ en", "en ⇒ zh", and "en ⇒ ar". Our method LAPIN and X-GEAR are both based on the mT5-base.

(2) Using the sequential selection strategy to initialize the prefix brings performance degradation. The reason is that this strategy ignores the triggers, and dependencies between triggers and candidate arguments. It suggests that making full use of trigger information is important for the task. In addition, compared with the model removed prefix-tuning, our method with the sequential selection still achieves better performance. It demonstrates that language-oriented prefix-tuning is very effective for the task.

Discussion and Analysis

Sensitivity to the Distance between Triggers and Arguments The distance between triggers and candidate arguments is very different among languages (e.g., 9.8 in English *vs* 21.7 in Chinese). Intuitively, if an EAE model is less sensitive to the distance between triggers and arguments, the model can be assumed to less overfit the source language. Table 4 shows the results on the ACE-2005 dataset with different distance between triggers and arguments. From the results, we can observe that our method LAPIN outperforms the baseline X-GEAR on almost all distance distributions. More importantly, compared with X-GEAR, our method achieves greater improvement when the distance between triggers and arguments becomes longer. It suggests that our method can handle discrepancies between source and target languages and avoid overfitting the source language.

Impact of Language-oriented Prefix Length We investigate the influence of language-oriented prefix length on the ACE-2005 dataset. The prefix length varies from 10 to 60, and the corresponding results are illustrated in Figure 4. From the figure, we can observe that the performance of our method LAPIN improves with the increase of prefix length at the beginning. Our method yields the best performance when the prefix length is set to 30. However, when the prefix length becomes too large, F1-score stops increasing or even decreases. We attribute it to the fact that dependency structures can shorten the distance between event triggers and their arguments and effectively model the long-distance dependencies between them. Therefore, the length of the language-oriented prefix need not be too long.

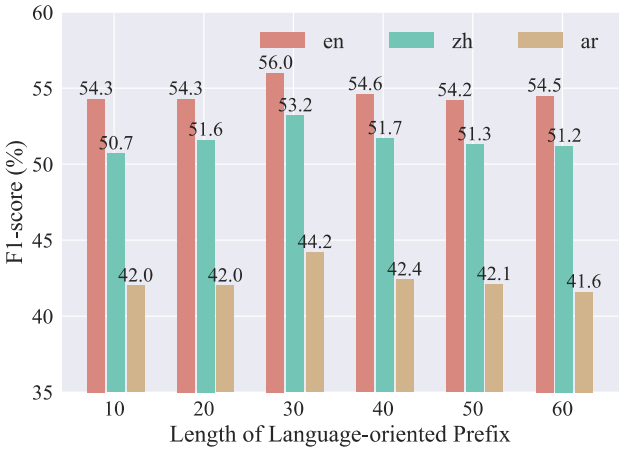

Figure 4: F1-scores of our method LAPIN with different prefix length on the ACE-2005. The "en" indicates the performance average of "en ⇒ en", "en ⇒ zh", and "en ⇒ ar".

Models	en ⇓ xx	ar ⇓ xx	zh ⇓ xx	avg
LAPIN (mT5-base)	**56.0**	**44.2**	**53.2**	**51.1**
w/o copy mechanism	54.8	43.2	51.1	49.7 (↓1.4)
LAPIN (mT5-large)	**61.9**	**50.2**	**56.3**	**56.1**
w/o copy mechanism	58.0	48.2	56.0	54.1 (↓2.0)

Table 5: Ablation study of copy mechanism on the ACE-2005 dataset. The "avg" denotes the performance average of "en ⇒ xx", "ar ⇒ xx", and "zh ⇒ xx".

Impact of Copy Mechanism To verify the effectiveness of the copy mechanism, we conduct ablation studies. The experimental results are listed in Table 5. From the table, we can observe that removing the copy mechanism brings performance degradation in all cross-lingual settings. Compared with the model removed copy mechanism, our methods LAPIN (mT5-base) and LAPIN (mT5-large) achieves 1.4% and 2.0% improvements of average F1-score, respectively. It suggests that the generative PLMs (i.e., mT5) lack the ability to copy input, and the copy mechanism can facilitate the cross-lingual adaptation.

Conclusion

In this paper, we propose a novel language-oriented prefix-tuning network (LAPIN) for zero-shot cross-lingual event argument extraction. To handle discrepancies between source and target languages, we devise a language-oriented prefix generator module to obtain prefixes based on language-universal dependency structures. Moreover, we leverage a language-agnostic template constructor module to design universal templates for facilitating cross-lingual transfer. Experimental results on two datasets indicate that our approach substantially outperforms previous state-of-the-art methods. In the future, we plan to adapt our method to other zero-shot cross-lingual information extraction tasks.

Acknowledgments

We thank anonymous reviewers for their insightful comments and suggestions. This work is supported by the National Key Research and Development Program of China (No.2020AAA0106400), the National Natural Science Foundation of China (No.62176257, 61976211, 61922085). This work is also supported by the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No.XDA27020200), the Youth Innovation Promotion Association CAS, and Yunnan Provincial Major Science and Technology Special Plan Projects (No.202103AA080015).

References

Ahmad, W. U.; Peng, N.; and Chang, K.-W. 2021. GATE: graph attention transformer encoder for cross-lingual relation and event extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 12462–12470.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 1877–1901.

Chen, Y.; Xu, L.; Liu, K.; Zeng, D.; and Zhao, J. 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*, 167–176.

Chen, Z.; and Ji, H. 2009. Can one language bootstrap the other: a case study on event extraction. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, 66–74.

Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, É.; Ott, M.; Zettlemoyer, L.; and Stoyanov, V. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8440–8451.

Doddington, G. R.; Mitchell, A.; Przybocki, M. A.; Ramshaw, L. A.; Strassel, S. M.; and Weischedel, R. M. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, 837–840.

Du, X.; and Cardie, C. 2020. Event Extraction by Answering (Almost) Natural Questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 671–683.

Han, X.; Zhao, W.; Ding, N.; Liu, Z.; and Sun, M. 2022. Ptr: Prompt tuning with rules for text classification. *AI Open*.

Hsi, A.; Yang, Y.; Carbonell, J. G.; and Xu, R. 2016. Leveraging multilingual training for limited resource event extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 1201–1210.

Hsu, I.-H.; Huang, K.-H.; Boschee, E.; Miller, S.; Natarajan, P.; Chang, K.-W.; and Peng, N. 2022. DEGREE: A Data-Efficient Generation-Based Event Extraction Model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1890–1908.

Huang, K.-H.; Hsu, I.-H.; Natarajan, P.; Chang, K.-W.; and Peng, N. 2022. Multilingual Generative Language Models for Zero-Shot Cross-Lingual Event Argument Extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 4633–4646.

Huang, L.; Ji, H.; Cho, K.; Dagan, I.; Riedel, S.; and Voss, C. 2018. Zero-Shot Transfer Learning for Event Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2160–2170.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045–3059.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 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*, 7871–7880.

Li, F.; Peng, W.; Chen, Y.; Wang, Q.; Pan, L.; Lyu, Y.; and Zhu, Y. 2020a. Event extraction as multi-turn question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 829–838.

Li, M.; Ma, T.; Yu, M.; Wu, L.; Gao, T.; Ji, H.; and McKeown, K. 2021. Timeline summarization based on event graph compression via time-aware optimal transport. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6443–6456.

Li, M.; Zareian, A.; Lin, Y.; Pan, X.; Whitehead, S.; Chen, B.; Wu, B.; Ji, H.; Chang, S.-F.; Voss, C.; et al. 2020b. Gaia: A fine-grained multimedia knowledge extraction system. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 77–86.

Li, S.; Ji, H.; and Han, J. 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*, 894–908.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 4582–4597.

Lin, Y.; Ji, H.; Huang, F.; and Wu, L. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, 7999–8009.

Liu, J.; Chen, Y.; Liu, K.; Bi, W.; and Liu, X. 2020. Event extraction as machine reading comprehension. In *Proceedings*

of the 2020 Conference on Empirical Methods in Natural Language Processing, 1641–1651.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Liu, X.; Huang, H.-Y.; Shi, G.; and Wang, B. 2022. Dynamic Prefix-Tuning for Generative Template-based Event Extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 5216–5228.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lou, C.; Gao, J.; Yu, C.; Wang, W.; Zhao, H.; Tu, W.; and Xu, R. 2022. Translation-based implicit annotation projection for zero-shot cross-lingual event argument extraction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2076–2081.

Ma, Y.; Wang, Z.; Cao, Y.; Li, M.; Chen, M.; Wang, K.; and Shao, J. 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*, 6759–6774.

Nguyen, T. H.; Cho, K.; and Grishman, R. 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*, 300–309.

Paolini, G.; Athiwaratkun, B.; Krone, J.; Ma, J.; Achille, A.; Anubhai, R.; dos Santos, C. N.; Xiang, B.; and Soatto, S. 2021. Structured Prediction as Translation between Augmented Natural Languages. In *9th International Conference on Learning Representations, ICLR 2021*.

Qi, P.; Zhang, Y.; Zhang, Y.; Bolton, J.; and Manning, C. D. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 101–108.

Qin, G.; and Eisner, J. 2021. Learning How to Ask: Querying LMs with Mixtures of Soft Prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 5203–5212.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 1–67.

Schick, T.; and Schütze, H. 2021. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, 255–269.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 1073–1083.

Seoh, R.; Birle, I.; Tak, M.; Chang, H.-S.; Pinette, B.; and Hough, A. 2021. Open Aspect Target Sentiment Classification with Natural Language Prompts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6311–6322.

Song, Z.; Bies, A.; Strassel, S.; Riese, T.; Mott, J.; Ellis, J.; Wright, J.; Kulick, S.; Ryant, N.; and Ma, X. 2015. From light to rich ere: annotation of entities, relations, and events. In *Proceedings of the the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, 89–98.

Subburathinam, A.; Lu, D.; Ji, H.; May, J.; Chang, S.-F.; Sil, A.; and Voss, C. 2019. Cross-lingual structure transfer for relation and event extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 313–325.

Van Nguyen, M.; and Nguyen, T. H. 2021. Improving cross-lingual transfer for event argument extraction with language-universal sentence structures. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, 237–243.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*.

Wang, X.; Wang, Z.; Han, X.; Liu, Z.; Li, J.; Li, P.; Sun, M.; Zhou, J.; and Ren, X. 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*, 5777–5783.

Xue, L.; Constant, N.; Roberts, A.; Kale, M.; Al-Rfou, R.; Siddhant, A.; Barua, A.; and Raffel, C. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 483–498.

Zhang, T.; Chen, M.; and Bui, A. A. 2020. Diagnostic prediction with sequence-of-sets representation learning for clinical events. In *International Conference on Artificial Intelligence in Medicine*, 348–358.

Zhu, Z.; Li, S.; Zhou, G.; and Xia, R. 2014. Bilingual event extraction: a case study on trigger type determination. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 842–847.