Multilingual Generative Language Models for Zero-Shot Cross-Lingual Event Argument Extraction

Anonymous ACL submission

Abstract

We present a pioneering study on leveraging multilingual pre-trained *generative* language models for zero-shot cross-lingual event argument extraction (EAE) by formulating EAE as a *language generation* task. Compared to previous classification-based EAE models that build classifiers on top of pre-trained *masked* language models, our generative model effectively encodes the event structures and better captures the dependencies between arguments. To achieve cross-lingual transfer, we design *language-agnostic templates* to encode argument roles, and train our models on source languages to "generate" arguments in the source languages to fill in the language-agnostic template. The trained model can then be directly applied to target languages to "generate" arguments in the target languages to fill in the template. Our experimental results demonstrate that the proposed model outperforms the current state-of-the-art results on zero-shot cross-lingual EAE. Comprehensive ablation study and error analysis are presented to better understand the advantages and the current limitations of using multilingual generative language models for cross-lingual transfer.

1 Introduction

Event argument extraction (EAE) aims to recognize the entities serving as event arguments and identify their corresponding roles. As illustrated by the English example in Figure 1, given a trigger word *"destroyed"* for a *Conflict:Attack* event, an event argument extractor is expected to identify *"commando"*, *"Iraq"*, and *"post"* as the event arguments and predict their corresponding roles.

Zero-shot cross-lingual EAE has attracted considerable attention since it eliminates the requirement of labeled data for constructing EAE models in low-resource languages (Subburathinam et al., 2019; Ahmad et al., 2021; Nguyen and Nguyen, 2021). In this setting, the model is trained on examples from the *source* languages and directly tested on instances from the *target* languages.

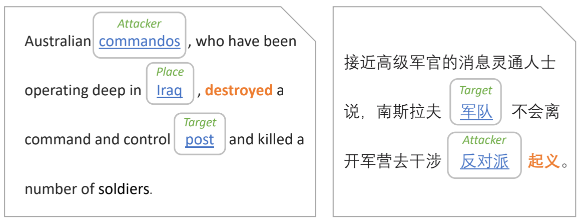


Figure 1: An illustration of cross-lingual event argument extraction. Given sentences in arbitrary languages and their event triggers (*destroyed* and 起义), the model needs to identify arguments (*commando*, *Iraq* and *post* v.s. 军队, and 反对派) and their corresponding roles.

Recently, pre-trained *generative* language models have shown strong performances on monolingual structured prediction tasks (Yan et al., 2021; Huang et al., 2021; Paolini et al., 2021) including the EAE task (Li et al., 2021; Hsu et al., 2021). These works treat structured prediction problems as language generation tasks and fine-tune pre-trained generative language models to generate outputs following designed templates such that the final predictions can be easily decoded from the outputs. They better capture the structures and dependencies between entities comparing to the traditional classification-based models (Wang et al., 2019; Lin et al., 2020) as the templates provide additional declarative information.

Despite the successes, the designs of templates in prior works are language-dependent, which makes it hard to extend them to the zero-shot cross-lingual transfer setting. Naively applying such models trained on the source languages to the target languages usually generates *code-switching* outputs, yielding poor performance for zero-shot cross-lingual transfer.[1] How to design *language-agnostic* generative models for zero-shot cross-lingual structured prediction problems is still an open question.

[1]We will show this empirically in Section 6.

In this work, we present a pioneering study of leveraging multilingual pre-trained generative models for zero-shot cross-lingual event argument extraction and propose X-GEAR (a **Cross**-lingual **G**enerative **E**vent **A**rgument extracto**r**).X-GEAR enables the knowledge transfer across languages by using *language-agnostic templates*, which serve as unified media that can carry arguments information in different languages. Given an input passage and a carefully designed prompt that contains an event trigger and the language-agnostic template, X-GEAR is trained to generate a sentence to fill in the language-agnostic template with arguments. X-GEAR inherits the strength of generative models — it captures the event structures and the dependencies between entities better than classification-based models. In addition, the pre-trained decoder inherently identifies named entities as candidates for event arguments, and does not need an additional module for named entity recognition.

We conduct experiments on two multilingual EAE datasets: ACE-2005 (Doddington et al., 2004) and ERE (Song et al., 2015). The results demonstrate that X-GEAR outperforms the state-of-the-art zero-shot EAE models. We further perform ablation studies to justify our design and present comprehensive error analysis to understand the limitations of using multilingual generative language models for zero-shot cross-lingual transfer.

2 Related Work

Zero-shot cross-lingual structured prediction. Zero-shot cross-lingual learning becomes an emerging research topic as it eliminates the requirement of labeled data for training models in low-resource languages. Various structured prediction tasks have be studied, including named entity recognition (Pan et al., 2017), dependency parsing (Ahmad et al., 2019), relation extraction (Zou et al., 2018; Ni and Florian, 2019), event detection (Huang et al., 2018; Liu et al., 2019), and event argument extraction (Subburathinam et al., 2019; Ahmad et al., 2021; Nguyen and Nguyen, 2021). Most of them are *classification-based models* that build classifiers on top of a multilingual pre-trained *masked* language models. To further deal with the discrepancy between languages, some of them require additional information, such as bilingual dictionaries (Liu et al., 2019; Ni and Florian, 2019), translation pairs (Zou et al., 2018), and dependency parse trees (Subburathinam et al., 2019; Ahmad et al., 2021; Nguyen and Nguyen, 2021). However, as pointed out by previous literature (Li et al., 2021; Hsu et al., 2021), classification-based models are less flexible to include the structure information and less powerful to model dependencies between entities compared to *generation-based models*.

Generation-based structured prediction. Several works have demonstrated that pre-trained *generative* language models lead to impressive performance on monolingual structured prediction tasks, including named entity recognition (Yan et al., 2021), relation extraction (Huang et al., 2021; Paolini et al., 2021), and event extraction (Du et al., 2021; Li et al., 2021; Huang et al., 2021; Hsu et al., 2021; Lu et al., 2021). Nevertheless, as mentioned in Section 1, their designed generating targets are language-dependent. Accordingly, directly applying their methods to the zero-shot cross-lingual setting would result in bad performance.

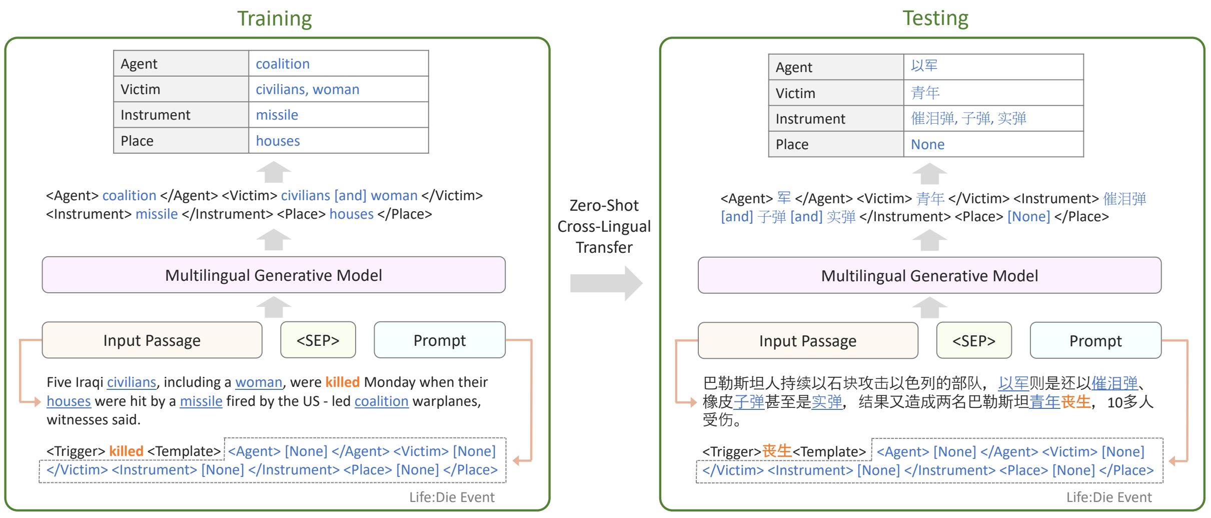
3 Zero-Shot Cross-Lingual Event Argument Extraction

In this paper, we focus on the zero-shot cross-lingual EAE. Given an input passage and an event trigger, an EAE model identifies arguments and their corresponding roles. More specifically, as illustrated by the training examples in Figure 2, given an input passage $\mathbf{x}$ and an event trigger $\mathbf{t}$ (*killed*) that belongs to event type $\mathbf{c}$ (*Life:Die*), an EAE model predicts a list of arguments $\mathbf{a} = [a_1, a_2, ..., a_l]$ (*coalition*, *civilians*, *woman*, *missile*, *houses*) and their corresponding roles $\mathbf{r} = [r_1, r_2, .., r_l]$ (*Agent*, *Victim*, *Victim*, *Instrument*, *Place*). In a zero-shot cross-lingual setting, the training instances $X_{train} = \{(\mathbf{x}_i, \mathbf{t}_i, \mathbf{c}_i, \mathbf{a}_i, \mathbf{r}_i)\}_{i=1}^{N}$ belong to some source languages while the testing instances $X_{test} = \{(\mathbf{x}_i, \mathbf{t}_i, \mathbf{c}_i, \mathbf{a}_i, \mathbf{r}_i)\}_{i=1}^{M}$ are in other languages.

Similar to monolingual EAE, zero-shot cross-lingual EAE models are expected to capture the dependencies between arguments and make structured predictions accordingly. However, unlike monolingual EAE, a zero-shot cross-lingual EAE model has to overcome the differences between languages (e.g., grammars, word orders) and learn to transfer the knowledge from the source languages to the target languages.

4 Proposed Method

We formulate zero-shot cross-lingual EAE as a language generation task and propose X-GEAR



Training
Agent
coalition
Victim
civilians, woman
Instrument
missile
Place
houses
<Agent> coalition </Agent> <Victim> civilians [and] woman </Victim> <Instrument> missile </Instrument> <Place> houses </Place>
Multilingual Generative Model
Input Passage
<SEP>
Prompt
Five Iraqi civilians, including a woman, were killed Monday when their houses were hit by a missile fired by the US - led coalition warplanes, witnesses said.
<Trigger> killed <Template> <Agent> [None] </Agent> <Victim> [None] </Victim> <Instrument> [None] </Instrument> <Place> [None] </Place>
Life:Die Event
Zero-Shot Cross-Lingual Transfer
Testing
Agent
以军
Victim
青年
Instrument
催泪弹, 子弹, 实弹
Place
None
<Agent> 军 </Agent> <Victim> 青年 </Victim> <Instrument> 催泪弹 [and] 子弹 [and] 实弹 </Instrument> <Place> [None] </Place>
Multilingual Generative Model
Input Passage
<SEP>
Prompt
巴勒斯坦人持续以石块攻击以色列的部队，以军则是还以催泪弹、橡皮子弹甚至是实弹，结果又造成两名巴勒斯坦青年丧生，10多人受伤。
<Trigger>丧生<Template> <Agent> [None] </Agent> <Victim> [None] </Victim> <Instrument> [None] </Instrument> <Place> [None] </Place>
Life:Die Event

Figure 2: The overview of X-GEAR. Given an input passage and a carefully designed prompt containing an event trigger and the language-agnostic template, X-GEAR fills in the language-agnostic template with event arguments.

(**Cross**-lingual **G**enerative **E**vent **A**rgument extracto**r**), which is depicted in Figure 2. There are two challenges raised by this formulation: (1) The generated output string needs to be easily decoded into final predictions. (2) The input language may vary during training and testing; therefore, the output strings have to reflect the change of the input language accordingly while remaining well-structured so the first point still holds.

We address these challenges by designing *language-agnostic templates*. Specifically, given an input sentence $\mathbf{x}$ and a designed prompt that encodes the trigger $\mathbf{t}$, its event type $\mathbf{c}$, and other auxiliary information, X-GEAR generates an output string following a language-agnostic template. The language-agnostic template is designed to be decoded easily so that the process of extracting the final argument predictions $\mathbf{a}$ and role predictions $\mathbf{r}$ from the generated output string can be easily executed. Moreover, since the template is language-agnostic, our method works regardless of the input language.

X-GEAR fine-tunes a multilingual pre-trained generative model, such as mBART-50 (Tang et al., 2020) or mT5 (Xue et al., 2021), while it is augmented with copy mechanism to better adapt to the input language change. In the following sections, we present the details of X-GEAR, including the language-agnostic templates, the target output string, and the input format.

4.1 Language-Agnostic Template

We create a language-agnostic template $T_{\mathbf{c}}$ for every event type $\mathbf{c}$. For each event type, we list all of its possible associated roles[2] and form a unique HTML-tag-style template for that event type $\mathbf{c}$. More precisely, in Figure 2, the *Life:Die* event is associated with four roles: *Agent*, *Victim*, *Instrument*, and *Place*. As a result, the template of *Life:Die* event is designed as:

<Agent>[None]</Agent><Victim>[None]</Victim>
<Instrument>[None]</Instrument><Place>[None]</Place>.

For the ease of understanding, we use English words to present the template. However, these tokens ([None], <Agent>, </Agent>, <Victim>, etc.) are encoded as special tokens[3] that the pre-trained models have never seen and thus their representations need to be learned from scratch. Since these special tokens are not associate with any language and are not pre-trained, they are considered as *language-agnostic*.

4.2 Target Output String

X-GEAR learns to generate target output strings that follow the form of language-agnostic templates. Given an example $(\mathbf{x}, \mathbf{t}, \mathbf{c}, \mathbf{a}, \mathbf{r})$, we first pick out the language-agnostic template $T_{\mathbf{c}}$ for the event type $\mathbf{c}$ and then replace all "[None]" in $T_{\mathbf{c}}$ with the corresponding arguments in $\mathbf{a}$ according to their roles $\mathbf{r}$. If there are multiple arguments for one role, we concatenate them with a special token "[and]". For instance, the

[2]The associated roles can be obtained by skimming training data or directly from the annotation guideline if provided.

[3]In fact , the special tokens can be replaced by any other format, such as <–token1–> or </–token1–>. Here, we use <Agent> and </Agent> to highlight that arguments between these two special tokens are corresponding to the *Agent* role.

training example in Figure 2 has two arguments (*civilians* and *woman*) for the *Victim* role, and the corresponding part of the output string would be

```
<Victim> civilians [and] woman </Victim>.
```

By applying this rule, the full output string for the training example in Figure 2 becomes

```
<Agent> coalition </Agent><Victim> civilians[and]
woman </Victim><Instrument> missile </Instrument>
<Place> houses </Place>.
```

Since the output string is in the HTML-tag style, we can easily decode the argument and role predictions from the generated output string via a simple rule-based algorithm.

4.3 Input Format

As we mentioned previously, the key for the generative formulation of zero-shot cross-lingual EAE is to guide the model to generate output strings in a desired format. To facilitate this behavior, we instruct X-GEAR by feeding a *prompt* together with input sentence $\mathbf{x}$ to it, as shown by Figure 2. A *prompt* contains all valuable information for the model to make a prediction, including a trigger $\mathbf{t}$ and a language-agnostic template $T_{\mathbf{c}}$. Notice that we do not *explicitly* include the event type $\mathbf{c}$ in the prompt because the template $T_{\mathbf{c}}$ *implicitly* contains this information. Later on, in Section 6.2, we will demonstrate the experiments on explicitly adding event type $\mathbf{c}$ to the prompt and discuss about how it influences the cross-lingual transferability.

4.4 Training

To enable X-GEAR to generate sentences in different languages, we resort multilingual pre-trained generative model to be our base model, which models the conditional probability of generating a new token given the previous generated tokens and the input context to the encoder c, i.e,

$$P(x|c) = \prod_i P_{gen}(x_i|x_{<i}, c),$$

where x_i is the output of the decoder at step i.

Copy mechanism. Although the multilingual pre-trained generative models can generate sequences in many languages, fully relying on them may results in generating hallucinating arguments (Li et al., 2021). Observing that most of the tokens in the target output string appear in the input sequence[4], we augment the multilingual pre-trained generative models with copy mechanism to help X-GEAR better adapt to the cross-lingual scenario. Specifically, we follow See et al. (2017) to decide the conditional probability of generating a token t as a weighted sum of the vocabulary distribution computed by multilingual pre-trained generative model P_{gen} and copy distribution P_{copy}

$$P_{\text{X-GEAR}}(x_i = t|x_{<i}, c) = w_{copy} \cdot P_{copy}(t) + (1 - w_{copy}) \cdot P_{gen}(x_i = t|x_{<i}, c)$$

where $w_{copy} \in [0, 1]$ is the copy probability computed by passing the decoder hidden state at time step i to a linear layer. As for P_{copy}, it refers to the probability over input tokens[5] weighted by the cross-attention that the last decoder layer computed (at time step i). Our model is then trained end-to-end with the following loss:

$$\mathcal{L} = -\log \sum_i P_{\text{X-GEAR}}(x_i|x_{<i}, c).$$

5 Experimental Settings

5.1 Datasets

We consider two commonly used event extraction datasets: ACE-2005 and ERE. **ACE-2005** (Doddington et al., 2004) provides entities, value, time, relation, and event annotations for English, Arabic, and Chinese. We follow the pre-processing in Wadden et al. (2019) and keep 33 event types and 22 argument roles. For the English split, we use the split provided by Wadden et al. (2019). For the Chinese portion, we follow the setting in Lin et al. (2020). As for the Arabic part, we adopt the setup proposed by Xu et al. (2021). Observing that part of the sentence breaks made from Xu et al. (2021) being extremely long for pretrained models to encode, we perform additional preprocessing and postprocessing procedures for Arabic data. Specifically, we split Arabic sentences into several portions that any of the portion is shorter than 80 tokens. Then, we map the models' predictions of the split sentences back to the original sentence during postprocessing.

ERE (Song et al., 2015) is created by the Deep Exploration and Filtering of Test (DEFT) program. We consider its English and Spanish annotations and follow the pre-processing in Lin et al. (2020), which keeps 38 event types and 21 argument roles for both languages. Table1 shows more details about the two datasets.

[4]Except for the special token `[and]`.

[5]If t does not appear in the input sequence, then the probability becomes zero.

Dataset	Lang.	Train #Events	Train #Args	Dev #Events	Dev #Args	Test #Events	Test #Args
ACE-2005	en	4202	4859	450	605	403	576
	ar	1743	2506	117	174	198	287
	zh	2926	5581	217	404	190	336
ERE	en	6208	8924	525	730	551	882
	es	3131	4415	204	279	255	354

Table 1: Dataset statistics of ACE-2005 and ERE.

Notice that prior works working on zero-shot cross-lingual transfer of event arguments mostly focus on event argument role labeling (Subburathinam et al., 2019; Ahmad et al., 2021), where they assume ground truth entities are provided during both training and testing. In their experimental data splits, events in a sentence can be scattered in all training, development, and test split since they treat each event-entity pair as a different instance. In this work, we consider event argument extraction (Wang et al., 2019; Wadden et al., 2019; Lin et al., 2020), which is a more realistic setting.

5.2 Evaluation

We follow previous work (Lin et al., 2020; Ahmad et al., 2021) and consider the *argument classification F1 score* to measure the performance of models. An argument-role pair is counted as correct if both the argument offsets and the role type match the ground truth. Given the ground truth arguments $\mathbf{a}$, ground truth roles $\mathbf{r}$, predicted arguments $\tilde{\mathbf{a}}$, and predicted roles $\tilde{\mathbf{r}}$, the argument classification F1 score is defined as the F1 score between the set $\{(\mathbf{a}_i, \mathbf{r}_i)\}$ and the set $\{(\tilde{\mathbf{a}}_j, \tilde{\mathbf{r}}_j)\}$. For every model, we experiment it with three different random seeds and report the average results.

5.3 Compared Models

We compare the following models and their implementation details are listed in Appendix A.

- **OneIE** (Lin et al., 2020), the state-of-the-art for monolingual event extraction, is a classification-based model trained with multitasking, including entity extraction, relation extraction, event extraction, and *event argument extraction*. We simply replace its pre-trained embedding with XLM-RoBERTa-large (Conneau et al., 2020) to fit the zero-shot cross-lingual setting. Note that the multi-task learning makes OneIE require *additional annotations*, such as named entity annotations and relation annotations.

- **CL-GCN** (Subburathinam et al., 2019) is a classification-based model for cross-lingual event argument role labeling (EARL). It considers *dependency parsing annotations* to bridge different languages and use GCN layers (Kipf and Welling, 2017) to encode the parsing information. We follow the implementation of previous work (Ahmad et al., 2021) and add two GCN layers on top of XLM-RoBERTa-large. Since CL-GCN focuses on EARL tasks, which assume the ground truth entities are available during testing, we add one name entity recognition module jointly trained with CL-GCN.

- **GATE** (Ahmad et al., 2021), the state-of-the-art model for zero-shot cross-lingual EARL, is a classification-based model which considers *dependency parsing annotations* as well. Unlike CL-GCN, it uses a Transformer layer (Vaswani et al., 2017) with modified attention to encode the parsing information. We follow the original implementation and add two GATE layers on top of XLM-RoBERTa-large. Similar to CL-GCN, we add one name entity recognition module jointly trained with GATE.

- **TANL** (Paolini et al., 2021) is a generative model for monolingual EAE. Their predicted target, augmented language, embeds labels into the input passage by using brackets and vertical bar symbols to mark the argument tokens with their corresponding label. To adapt TANL to zero-shot cross-lingual EAE, we replace its pre-trained based model T5 (Raffel et al., 2020) with mT5-base (Xue et al., 2021).

- **X-GEAR** is our proposed model. We consider three different pre-trained generative language models: mBART-50-large (Tang et al., 2020), mT5-base, and mT5-large (Xue et al., 2021).

6 Experimental Results

6.1 Main Results

Table 2 and Table 3 list the results on ACE-2005 and ERE, respectively, with all combinations of source languages and target languages. Note that all the models have similar numbers of parameters except for X-GEAR with mT5-large.

Comparison to prior generative models. We first observe that TANL has poor performance when transferring to different languages. The reason is that its language-dependent template makes

Model	# of parameters	en ⇓ en	en ⇓ zh	en ⇓ ar	ar ⇓ ar	ar ⇓ en	ar ⇓ zh	zh ⇓ zh	zh ⇓ en	zh ⇓ ar	avg
OneIE (XLM-R-large) (Lin et al., 2020)	∼560M	63.6	42.5	37.5	57.8	27.5	<u>31.2</u>	<u>69.6</u>	51.5	31.1	45.8
CL-GCN (XLM-R-large) (Subburathinam et al., 2019)	∼570M	59.8	29.4	25.0	47.5	25.4	19.4	62.2	40.8	23.3	37.0
GATE (XLM-R-large) (Ahmad et al., 2021)	∼630M	67.0	49.2	<u>44.5</u>	59.6	27.6	26.3	**70.6**	46.7	**37.3**	47.6
TANL (mT5-base) (Paolini et al., 2021)	∼560M	59.1	38.6	29.7	50.1	18.3	16.9	65.2	33.3	18.3	36.6
X-GEAR (mBART-50-large)	∼610M	<u>68.6</u>	50.0	37.9	60.5	<u>30.8</u>	30.4	65.4	46.1	31.4	46.8
X-GEAR (mT5-base)	∼580M	67.9	<u>53.1</u>	42.0	<u>65.9</u>	28.5	30.4	69.4	<u>52.8</u>	32.0	<u>49.1</u>
X-GEAR (mT5-large)	∼1200M	**71.1**	**54.0**	**45.1**	**68.2**	**33.6**	**33.0**	68.9	**55.0**	<u>33.1</u>	**51.3**

Table 2: Average results of ACE-2005 with three different seeds. The best is in bold and the second best is underlined. "en ⇒ zh" denotes that models transfers from en to zh. X-GEAR outperforms other baselines.

Model	en ⇓ en	en ⇓ es	es ⇓ es	es ⇓ en	avg
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
TANL (mT5-base)	65.9	40.3	58.6	47.4	53.1
X-GEAR (mBART-50-large)	69.5	57.3	63.9	58.9	62.4
X-GEAR (mT5-base)	<u>69.8</u>	57.9	<u>66.1</u>	<u>59.0</u>	<u>63.2</u>
X-GEAR (mT5-large)	**72.9**	<u>59.7</u>	**67.4**	**64.1**	**66.0**

Table 3: Average results of ERE with three different seeds. The best is in bold and the second best is underlined. "en ⇒ es" denotes that models transfers from en to es. X-GEAR outperforms other baselines.

TANL easily generate code-switching outputs[6], which is a case that pretrained generative model rarely seen, leading to the poor performance. In contrast, X-GEAR considers the language-agnostic templates and achieve better performance for zero-shot cross-lingual transfer.

Comparison to classification models. X-GEAR with mT5-base outperforms OneIE, CL-GCN, and GATE on almost all the combinations of the source language and the target language. Especially, the improvement becomes much larger when the source language is Arabic. This suggests that language generation framework is indeed a promising approach for zero-shot cross-lingual EAE.

It is worth noting that OneIE, CL-GCN, and GATE require additional pipeline named entity recognition module to make predictions. Moreover, CL-GCN and GATE needs additional dependency parsing annotations to align the representations of different languages. On the contrary, X-GEAR is able to leverage the learned knowledge from the pre-trained generative models and therefore no additional modules or annotations are needed.

[6]The output text contains predictions written in the target language and templated tokens representing in the source language.

Comparison to different pre-trained generative language models. Interestingly, using mT5-base is more effective than using mBART-50-large, even if they have similar amount of parameters. We conjecture that the use of special tokens leads to this difference. mBART-50 has different begin-of-sequence (BOS) tokens for different languages. During generation, we have to specify which BOS token we would like to use as the start token. We guess that this language specific BOS token makes mBART-50 harder to transfer the knowledge from the source language to the target language. Unlike mBART-50, mT5 does not have such language specific BOS tokens. During generation, mT5 uses the padding token as the start token to generate sequence. This design is more general and benefit zero-shot cross-lingual transfer.

Larger pre-trained models are better. Finally, we demonstrate that the performance of X-GEAR can be further boosted with a larger pre-trained generative language models. As shown by Table 2 and Table3, X-GEAR with mT5-large achieves the best scores on most of the cases.

6.2 Ablation Study

Copy mechanism. We first study the effect of copy mechanism. Table 4 lists the performance of X-GEAR with and without copy mechanism. It shows improvements of adding copy mechanism when using mT5-large and mT-base. However, interestingly, adding copy mechanism is not effective to mBART-50. We conjecture that this is because the pre-trained objective of mBART-50 is denoising and reconstructing the original sentence (Liu et al., 2020), hence, mBART-50 has already learn to copy tokens from the input. Therefore, adding copy mechanism is less useful. On the other hand, the pre-trained objective of mT5 is various NLP tasks which are far away from copying input. The

Model	en ⇓ xx	ar ⇓ xx	zh ⇓ xx	xx ⇓ en	xx ⇓ ar	xx ⇓ zh	avg
mT5-large	**56.8**	44.9	**52.3**	**53.2**	**48.8**	52.0	**51.3**
- w/o copy	55.1	**45.0**	51.5	52.0	46.3	**53.2**	50.5
mT5-base	**54.3**	**41.6**	**51.4**	**49.7**	**46.6**	**51.0**	**49.1**
- w/o copy	52.1	39.5	47.6	48.1	42.7	48.5	46.4
mBART-50-large	**52.2**	40.6	47.6	48.5	43.3	48.6	46.8
- w/o copy	50.9	**42.2**	**49.6**	**50.6**	**43.5**	**48.7**	**47.6**

Table 4: Ablation study on copy mechanism for ACE-2005. “en ⇒ xx” indicates the average of “en ⇒ en”, “en ⇒ zh”, and “en ⇒ ar”.

copy mechanism thus becomes beneficial to mT5.

Including event type in prompts. In Section 4, we mentioned that the designed prompt for X-GEAR consists of only the input sentence and the language-agnostic template. In this section, we discuss whether explicitly including the event type information in the prompt is helpful. We consider three ways to include the event type information:

- **English tokens**. We put the English version of the event type in the prompt even if we are training or testing on non-English languages, for example, *Attack* stands for the event type *Attack*.
- **Translated tokens**. For each event type, we prepare the translated version of that event type token. For example, we use 攻击 to represent the *Attack* event type. During training or testing, we decide the language according to the language of the input passage. Notice that all the event types are written in English in the ACE-2005 and ERE. Hence, we use off-the-self machine translation tool to perform the translation.
- **Special tokens**. We create a special token for every event type and let the model to learn the representations of the special tokens from scratch. For instance, we use `<-attack->` to represent the *Attack* event type.

Table 5 shows the ablation study. In most cases, including event type information in the prompt drops the performance. One crucial reason is that one word in a language can be mapped to several words in another language. For example, the *Life* event type is related to *Marry*, *Divorce*, *Born*, and *Die* four sub-event types. In English, we can use just one word *Life* to cover all four sub-event types. However, In Chinese, when talking about *Marry* and *Divorce*, *Life* should be translated to “生活”; when talking about *Born* and *Die*, *Life* should be translated to “生命”. This mismatch causes the

Model	en ⇓ xx	ar ⇓ xx	zh ⇓ xx	xx ⇓ en	xx ⇓ ar	xx ⇓ zh	avg
X-GEAR	**54.3**	**41.6**	51.4	49.7	**46.6**	**51.0**	**49.1**
w/ English Tokens	53.3	39.3	**52.3**	49.2	46.5	49.2	48.3
w/ Translated Tokens	51.7	40.4	52.2	**49.8**	45.6	48.8	48.1
w/ Special Tokens	52.3	39.7	51.8	49.0	45.4	49.3	47.9

Table 5: Ablation study on including event type information in prompts for ACE-2005. “en ⇒ xx” indicates the average of “en ⇒ en”, “en ⇒ zh”, and “en ⇒ ar”.

performance drop when considering event types in prompts. Currently, we conclude that it is hard to utilize the information of event types in an appropriate way for all languages. How to resolve this challenge is considered as our future work.

7 Analysis

In this section, we perform error analysis of X-GEAR when transferring from Arabic to English and transferring from Chinese to English. For each case, we sample 30 failed examples and present the distribution of various error types in Figure 3. We discuss some of the categories as follows:

Errors on both monolingual and cross-lingual model. We compare the predicted results from X-GEAR(ar ⇒ en) with X-GEAR(en ⇒ en), or from X-GEAR(zh ⇒ en) with X-GEAR(en ⇒ en). If their predictions are similar and both of them are wrong when compared to the gold output, we classify the error to this category. To overcome the errors in this category, the potential solution is to improve monolingual models for EAE tasks.

Over generate. Errors in this category happen more often in X-GEAR(ar ⇒ en). It is likely because the entities in Arabic are usually much longer than that in English, when measuring by the number of sub-words. Based on our statistics, the average entity span length is 2.85 for Arabic, and is 2.00 for English (length of sub-words). This leads to the natural for our X-GEAR(ar ⇒ En) to overly generate some tokens even though they have captured the correct concept. An example is that the model predicts *“The EU foreign ministers”*, while the ground truth is *“ministers”*.

Label disagreement on different language split. The annotations for the ACE dataset in different language split contain some ambiguity. For example, given sentence *“He now also advocates letting in U.S. troops for a war against Iraq even though it is a fellow Muslim state.”* and the queried trigger

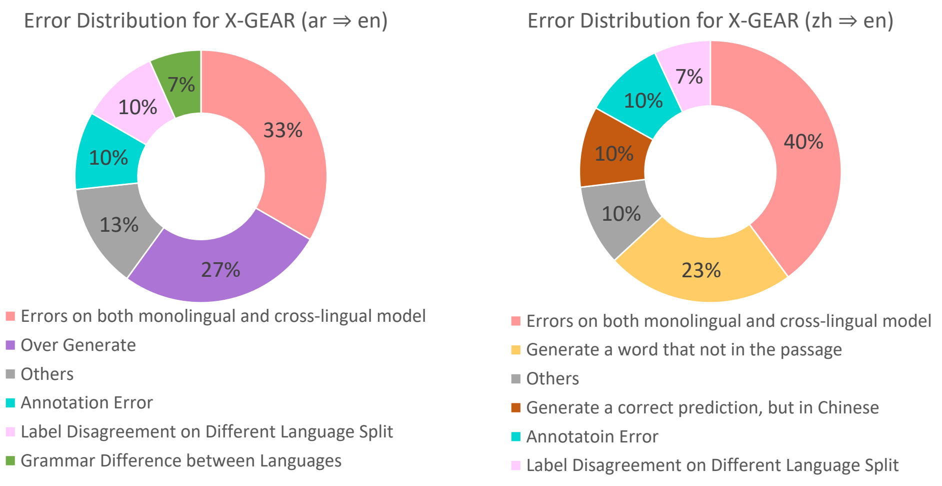


Figure 3: Distribution of errors that made by X-GEAR. ***Left:*** The distribution for our model that transfers from Arabic to English; ***Right:*** The distribution for our model trained on Chinese and tested on English.

"war", the annotations in English tends to label *Iraq* as the *Place* where the event happen, while similar situations in other languages will mark *Iraq* as the *Target* for the war.

Grammar difference between languages. An example for this category is *"... Blackstone Group would buy Vivendi's theme park division, including Universal Studios Hollywood ..."* and the queried trigger *"buy"*. We observe that X-GEAR(ar ⇒ en) predicts *Videndi* as the *Artifact* been sold and *division* is the *Seller*, while X-GEAR(en ⇒ en) can correctly understand that *Videndi* are the *Seller* and *division* is the *Artifact*. We hypothesize the reason being the differences between the grammar in Arabic and English. The word order of the sentence *"Vivendi's theme park division"* in Arabic is reversed with its English counterpart, that is, *"theme park division"* will be written before *"Vivendi"* in Arabic. Such difference leads to the errors in this category.

Generate a word that is not in the passage. In X-GEAR(zh ⇒ en), we observe several errors regarding generating a word that is not in the passage. There are two typical situations. The first case is that X-GEAR(zh ⇒ en) presents difficulty understanding singular and plural nouns. For example, the model will generate *"studios"* as prediction while only *"studio"* appear in the passage. This is because the concept regarding the differences between singular and plural nouns are less emphasized in Chinese. The second cases is that X-GEAR will generate some random predictions in Chinese, leading to false positives.

Generate a correct prediction, but in Chinese. This is a special case of "Generate a word that is not in the passage". In this category, we observe that although the prediction is in Chinese (hence, a wrong prediction), it is correct if we translate the prediction into English.

From these examples, we highlight two remaining challenges for future studies. First, there are several errors raising because of the discrepancies between the source language and the target language, such as the output length distribution mismatching or the grammar differences. This induces the research question on how we can mitigate this discrepancy without the help of using additional labels. Second, we demonstrate that even though we have already incorporated copy mechanism to facilitate the generation in target language, it is still challenging for the model to be fully controlled when adapting to cross-lingual cases. To cope with this issue, a potential solution is to use constrained decoding (Cao et al., 2021) to force all the generated tokens appearing in input. However, it is still an open question on how can we enforce this controlled generation more flexibly yet reliably.

8 Conclusion

We present a pioneering study on leveraging multilingual pre-trained generative language models for zero-shot cross-lingual event argument extraction. We design language-agnostic templates to overcome the discrepancy between languages and propose X-GEAR, a generative zero-shot cross-lingual event argument extractor. Our experimental results show that X-GEAR outperforms the current state-of-the-art models, which demonstrates the potential of using language generation framework to solve zero-shot cross-lingual structured prediction tasks.

References

Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. 2021. GATE: graph attention transformer encoder for cross-lingual relation and event extraction. In *Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*.

Wasi Uddin Ahmad, Zhisong Zhang, Xuezhe Ma, Eduard H. Hovy, Kai-Wei Chang, and Nanyun Peng. 2019. On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2021. Multilingual autoregressive entity linking. *arXiv preprint arXiv:2103.12528*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

George R. Doddington, Alexis Mitchell, Mark A. Przybocki, Lance A. Ramshaw, Stephanie M. Strassel, and Ralph M. Weischedel. 2004. The automatic content extraction (ACE) program - tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*.

Xinya Du, Alexander M. Rush, and Claire Cardie. 2021. GRIT: generative role-filler transformers for document-level event entity extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2021. Degree: A data-efficient generative event extraction model. *arXiv preprint arXiv:2108.12724*.

Kung-Hsiang Huang, Sam Tang, and Nanyun Peng. 2021. Document-level entity-based extraction as template generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare R. Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations (ICLR)*.

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 (NAACL-HLT)*.

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 (ACL)*.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2019. Neural cross-lingual event detection with minimal parallel resources. 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)*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguistics*, 8:726–742.

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 (ACL/IJCNLP)*.

Minh Van Nguyen and Thien Huu Nguyen. 2021. Improving cross-lingual transfer for event argument extraction with language-universal sentence structures. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*.

Jian Ni and Radu Florian. 2019. Neural cross-lingual relation extraction based on bilingual word embedding mapping. 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)*.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th*

International Conference on Learning Representations (ICLR).

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 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, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*.

Zhiyi Song, Ann Bies, Stephanie M. Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 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, (EVENTS@HLP-NAACL)*.

Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare R. Voss. 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 (EMNLP-IJCNLP)*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017 (NeurIPS)*.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh 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)*.

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)*.

Haoran Xu, Seth Ebner, Mahsa Yarmohammadi, Aaron Steven White, Benjamin Van Durme, and Kenton W. Murray. 2021. Gradual fine-tuning for low-resource domain adaptation. *arXiv preprint arXiv:2103.02205*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 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 (NAACL-HLT)*.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various NER subtasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL/IJCNLP)*.

Bowei Zou, Zengzhuang Xu, Yu Hong, and Guodong Zhou. 2018. Adversarial feature adaptation for cross-lingual relation classification. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*.

A Implementation Details

We describe the implementation details for all the models as follows:

- **OneIE** (Lin et al., 2020). We use their provided code[7] to train the model with the provided default settings. It is worth mention that for the Arabic split in the ACE-2005 dataset, OneIE is trained with only entity extraction, event extraction, and event argument extraction since there is no relation labels in Xu et al. (2021)'s preprocessing script. All other parameters are set to the default values.

- **CL-GCN** (Subburathinam et al., 2019). We refer the released code from Ahmad et al. (2021)[8] to re-implement the CL-GCN method. Specifically, we adapt the baseline framework that described and implemented in OneIE's code (Lin et al., 2020), but we remove its relation extraction module and add two layers of GCN on top of XLM-RoBERTa-large. The pos-tag and dependency parsing annotations are obtained by applying Stanza (Qi et al., 2020). All other parameters are set to the be the same as the training of OneIE.

- **GATE** (Ahmad et al., 2021). We refer the official released code from Ahmad et al. (2021)[9] to re-implement GATE. Similar to CL-GCN, we adapt the baseline framework that described and implemented in OneIE's code (Lin et al., 2020), but we remove its relation extraction module and add two layers of GATE on top of XLM-RoBERTa-large. The pos-tag and dependency parsing annotations are also obtained by applying Stanza (Qi et al., 2020). The hyperparameter of δ in GATE is set to be [2, 2, 4, 4, ∞, ∞, ∞, ∞]. All other parameters are set to the be the same as the training of OneIE.

- **TANL** (Paolini et al., 2021). To adapt TANL to zero-shot cross-lingual EAE, we adapt the public code[10] and replace its pre-trained based model T5 (Raffel et al., 2020) with mT5-base (Xue et al., 2021). All other parameters are set to their default values.

- **X-GEAR** is our proposed model. We consider three different pre-trained generative language models: mBART-50-large (Tang et al., 2020), mT5-base, and mT5-large (Xue et al., 2021). When fine-tune the pre-trained models, we set the learning rate to 10^{-4}. The batch size is set to 8. The number of epochs is 60.

[7] http://blender.cs.illinois.edu/software/oneie/

[8] https://github.com/wasiahmad/GATE

[9] https://github.com/wasiahmad/GATE

[10] https://github.com/amazon-research/tanl