Global Constraints with Prompting for Zero-Shot Event Argument Classification

Zizheng Lin[1], **Hongming Zhang**[2] **& Yangqiu Song**[1]
[1]Department of Computer Science and Engineering, HKUST
[2]Tencent AI Lab, Seattle
`{zlinai, yqsong}@cse.ust.hk`; `hongmzhang@global.tencent.com`

Abstract

Determining the role of event arguments is a crucial subtask of event extraction. Most previous supervised models leverage costly annotations, which is not practical for open-domain applications. In this work, we propose to use global constraints with prompting to effectively tackles event argument classification without any annotation and task-specific training. Specifically, given an event and its associated passage, the model first creates several new passages by prefix prompts and cloze prompts, where prefix prompts indicate event type and trigger span, and cloze prompts connect each candidate role with the target argument span. Then, a pre-trained language model scores the new passages, making the initial prediction. Our novel prompt templates can easily adapt to all events and argument types without manual effort. Next, the model regularizes the prediction by global constraints exploiting cross-task, cross-argument, and cross-event relations. Extensive experiments demonstrate our model's effectiveness: it outperforms the best zero-shot baselines by 12.5% and 10.9% F1 on ACE and ERE with given argument spans and by 4.3% and 3.3% F1, respectively, without given argument spans. We have made our code publicly available.[1]

1 Introduction

Event Argument Classification[2] (EAC), finding the roles of event arguments, is an important and challenging event extraction sub-task. As shown in Figure 1, a "Transfer-Money" event whose trigger is "acquiring" has several argument spans (e.g., "Daily Planet"). By determining the role of these arguments (e.g., "Daily Planet" as "Beneficiary"), we

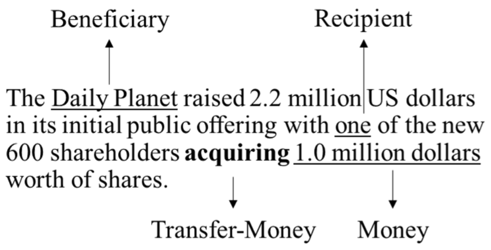

Beneficiary
Recipient
The Daily Planet raised 2.2 million US dollars in its initial public offering with one of the new 600 shareholders **acquiring** 1.0 million dollars worth of shares.
Transfer-Money
Money

Figure 1: An example of EAC. The trigger is in **bold face**. Arguments are underlined and connected to their roles by arrows.

can obtain a better understanding of the event, thus benefiting related applications like stock price prediction (Ding et al., 2015) and biomedical research (Zhao et al., 2021).

Many previous EAC works require numerous annotations to train their models (Lin et al., 2020; Hsu et al., 2022; Liu et al., 2022), which is not only costly as the annotations are labor-intensive but also difficult to be generalized to datasets of novel domains. Accordingly, some EAC models adopt a few-shot learning paradigm (Ma et al., 2022; Hsu et al., 2022). However, they are sensitive to the few-shot example selection and they still require costly task-specific training, which hinders their real-life deployment. There have been some zero-shot EAC models based on transfer learning (Huang et al., 2018), or label semantics (Zhang et al., 2021; Wang et al., 2022), or prompt learning (Liu et al., 2020; Lyu et al., 2021; Huang et al., 2022; Mehta et al., 2022). However, these models' corresponding limitations impede their real-life deployment. The model based on transfer learning can be ineffective when new event types are very different from the observed one. As for models using label semantics, they require a laborious preparation process and have unsatisfactory performance. Regarding models adopting prompt learning, they need tedious prompt design customized to every new type of events and arguments, and their performance is

[1]https://github.com/HKUST-KnowComp/Constraints-with-Prompting-for-Zero-Shot-EAC

[2]We focus on event argument because existing zero-shot trigger extraction models like Zhang et al. (2021) are already strong enough, but the arguments remain a challenge. Our argument identification approach is described in Section 3.1.

also limited.

To address the aforementioned issues, we propose an approach using global constraints with prompting to tackle zero-shot EAC. Global constraints can be viewed as a type of supervision signal from domain knowledge, which is crucial for zero-shot EAC since supervision from annotations is inaccessible. Moreover, our model's constraints module provides abundant global insights across tasks, arguments, and events. Prompting can also be regarded as a supervision signal as it induces abundant knowledge from Pre-Trained Language Models (PTLM). Unlike previous zero-shot EAC works, which need a tedious prompt design for every new type of events and arguments, the novel prompt templates of our model's prompting module can be easily adapted to all possible types of events and arguments in a fully automatic way. Specifically, given an event and its passage, our model first adds prefix prompt, cloze prompt, and candidate roles into the passage, which creates a set of new passages. The Prefix prompt describes the event type and trigger span. Cloze prompt connects each candidate to the target argument span. Afterwards, our model adopts a PTLM to compute the language modeling loss for each of the new passages, whose negative value would be the respective prompting score. The role with the highest prompting score is the initial prediction. Then, our model uses global constraints to regularize the initial prediction. The global constraints are based on the domain knowledge of the following relations: (1) cross-task relation, where our model additionally performs another one or more classification task on target argument span, and our model's predictions on EAC and other task(s) should be consistent; (2) cross-argument relation, where arguments of one event should collectively abide by certain constraint(s); (3) cross-event relation, where some argument playing a certain role in one event should play a typical role in another related event.

We conduct comprehensive experiments to demonstrate the effectiveness of our model. Particularly, our approach surpasses all zero-shot baselines by at least 12.5% and 10.9% F1 on ACE and ERE, respectively. When argument spans are not given, our model outperforms the best zero-shot baseline by 4.3% and 3.3% F1 on ACE and ERE, respectively. Besides that, we also conduct experiments to show that both the prompting and constraints modules contribute to the final success.

2 Methodology

We first present an overview of our approach. Then we introduce the details by describing its prompting module and global constraints regularization module. We follow (Liu et al., 2021) to name a prompt inserted before input text as *prefix prompt*, and a prompt with slot(s) to fill in and insert in the middle of input text as *cloze prompt*.

2.1 Overview

As shown in Figure 2, given a passage with a target argument span, our model infers the target's role without annotation and task-specific training. Our model has two modules. The first module is the prompting module that creates and scores several new passages. During creation, the model adds prefix prompt, cloze prompt, and candidate roles into the passage, where the prefix prompt contains information about event type and trigger, and the cloze prompt joins each candidate with a target argument span.[3] Afterwards, the model uses a PTLM to score the new passages. Our novel prompt templates can easily adapt to all possible events and arguments without manual work. Initial prediction is the role with the best prompting scores. The second module is the global constraints regularization module, where the model regularizes the prediction by three types of global constraints: cross-task constraint, cross-argument constraint, and cross-event constraint. All global constraints are based on event-related domain knowledge about inter-task, inter-argument, and inter-event relations.

2.2 Prompting Module

In this section, we describe the prompting module in detail. Given a passage, we first add a prefix prompt containing information about the event type and trigger span to the beginning. Such a prompt can guide a PTLM to: (1) accurately capture the input text's perspective related to the event; (2) have a clear awareness of the trigger. Based on the definitions of events and triggers (Grishman et al., 2005), we create the following prefix prompt: "**This is a [] event whose occurrence is most clearly expressed by [].**" where the first and second pairs of square brackets are the placeholders of event type and trigger span respectively. We also con-

[3]Since we focus on event argument classification, we assume that the event types and trigger spans are given. The settings without given argument spans will be discussed in Section 3.1.

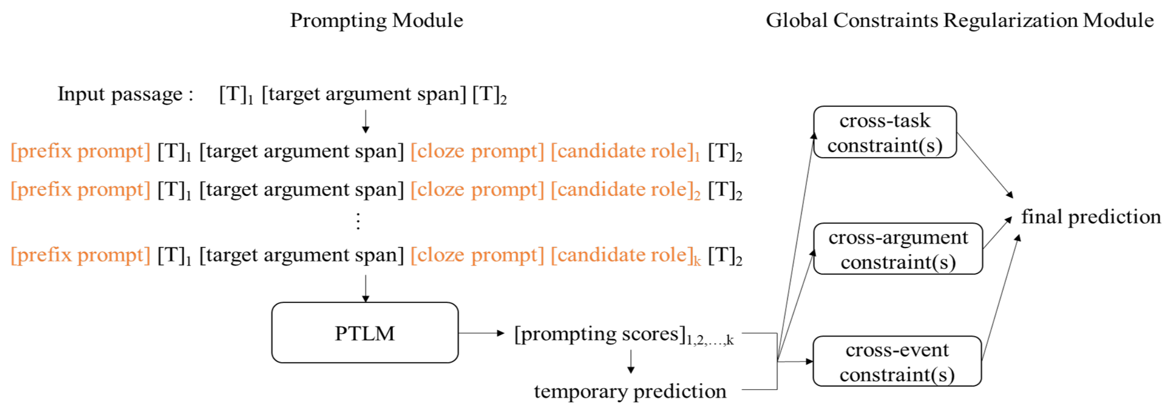

Prompting Module
Global Constraints Regularization Module
Input passage : [T]1 [target argument span] [T]2
[prefix prompt] [T]1 [target argument span] [cloze prompt] [candidate role]1 [T]2
[prefix prompt] [T]1 [target argument span] [cloze prompt] [candidate role]2 [T]2
[prefix prompt] [T]1 [target argument span] [cloze prompt] [candidate role]k [T]2
PTLM
[prompting scores]1,2,...,k
temporary prediction
cross-task constraint(s)
cross-argument constraint(s)
cross-event constraint(s)
final prediction

Figure 2: Model overview using prediction for one argument span as an example. $[T]_1$ and $[T]_2$ are the parts of the input passage before and after the span, respectively. k is the number of candidate roles of the event type.

ducted some experiments comparing different prefix prompts in Section A, and the results showed that the prefix above is the most effective.

Second, for each candidate role, the module inserts the cloze prompt behind the target argument span, and the role fills the prompt' slot. The cloze prompt adopts the hypernym extraction pattern "**M and any other []**" (Dai et al., 2021), where "M" denotes the argument span and the square bracket is the placeholder of the candidate role. We did not try other hypernym extraction patterns as (Dai et al., 2021) had shown that our pattern is the most effective. The motivation for adopting the hypernym extraction pattern for cloze prompt is that, to some extent a role can be regarded as a context-specific hypernym of the respective argument span of the associated event (e.g., "Beneficiary" can be seen as a context-specific hypernym of "Daily Planet"of the Transfer-Money event described by the example in Figure 1). Hence, such a prompt induces the linguistic and commonsense knowledge stored in PTLM to help identify which candidate role is the most reasonable.

After adding the previous two types of prompts, we get several new passages. For instance, suppose the passage is"In Baghdad, a bomb was fired at 17 people." whose event type is "Conflict:Attack", trigger is "fired", target argument span is "bomb", and candidate roles are {"Attacker", "Instrument", "Place", "Time", "Target"}. The created passages would be: (1) "*This is a Attack event whose occurrence is most clearly expressed by "fired."* In Baghdad, a bomb *and any other* *attacker* was fired at 17 people."; (2) "*This is a Attack ... "fired."* ... bomb *and any other* *instrument* was ..."; and similar text for other roles.[4]

For each new passage, we apply a PTLM to compute the language modeling loss. The negative value of the loss would be the prompting score of the respective passage, where a higher value indicates higher plausibility according to the PTLM. **Since our model's prompt templates are independent of event type and argument role, their adaptation to any new type of events and arguments is trivial and fully automatic.** Hence, our prompting method is more scalable and generalizable than those of previous zero-shot EAC models, since, for every new type of events and arguments they need to design a customized prompt. For instance, for every type of events/arguments, Lyu et al. (2021) manually design a unique prompt as text entailment/question answering template. The initial prediction would be the role with the highest prompting score. Since the steps of obtaining scores for each candidate role are independent of other candidate roles, we implement the steps of different candidate roles in parallel. Such a parallel implementation significantly improves our model's efficiency.

2.3 Global Constraints Regularization Module

This module regularizes the prediction by the following three types of global constraints.[5]

Cross-task constraint exploits the label dependency between EAC and auxiliary task(s) so that

[4]We only use the subtype of all events following the preprocessing done by (Lin et al., 2020)

[5]We designed 14 global constraints in total and we used preliminary experiments to choose the three most effective ones. In the preliminary experiments, we randomly sample 1k instances covering all trigger and argument types. We then evaluate each constraint on the sampled subset.

our model can get global information from the auxiliary task(s) about event arguments. We use **Event Argument Entity Typing (EAET)** as the auxiliary task. The task aims to classify an argument into its context-dependent entity type (e.g., PER). **As specified in ACE2005 ontology, an argument of a certain role in an event can only be one of several respective entity types (e.g., an argument of "Attack" role in a Conflict:Attack event can only be "ORG," "PER," or "GPE").** Based on this domain knowledge, we design the cross-task constraint as follows: (1) For each input passage, our model performs prompting for EAET, where the prompting is the same as in Section 2.2 except that candidate entity types replace the candidate roles in cloze prompt.; (2) After obtaining the scores and prediction of EAET, the model check the consistency between the predictions of EAC and EAET; (3) If the consistency is violated and the score of EAC's predicted role is lower, then discard the current role, use the role with the highest score in the remaining ones, and check the consistency again; (4) The constraint ends when the labels of two tasks are consistent. An example illustrating this type of constraint is shown in Figure 3.

Cross-argument constraint is based on domain knowledge about relationships between arguments within an event. Specifically, our model constrains the number of particular arguments for some or all events. For instance, it is very unlikely that an event mentioned is associated with multiple "Time" arguments. Such constraints offer a global understanding of event arguments to our model. The cross-argument constraint we adopt is "**A Personnel:End-POSITION event has at most one Position argument**." Given a Personnel:End-POSITION event, our model first checks the number of "Position" argument. If the number is more than one, then our model will first collect the arguments whose roles are "Position" and remove the one with the highest score among these arguments. Then for each remaining argument, our model would change the role to its candidate with the second highest score. An example illustrating this type of constraint is shown in Figure 4.

Cross-event constraint regularizes predicted roles of arguments shared by related events. A model with such a constraint can have global insights into event arguments, because while they are making inferences for the arguments of one event, they are aware of the information of other related event(s) and cross-event relations. The cross-event constraint we adopt is "**If a Life:Injure event and a Conflict:Attack event share arguments, then Injure.Place is the same as Attack.Place, Injure.Victim is the same as Attack.Target, Injure.Instrument is the same as Attack.Instrument, Injure.Time is the same as Attack.Time, Injure.Agent is the same as Attack.Attacker**". Given a passage containing an Injure and an Attack event sharing arguments, the model imposes the constraint by checking the consistency between the respective roles of each shared argument as specified in the constraint. Any inconsistency would be fixed by changing the role with a lower prompting score to the new one satisfying the consistency. An example illustrating this type of constraint is shown in Figure 5.

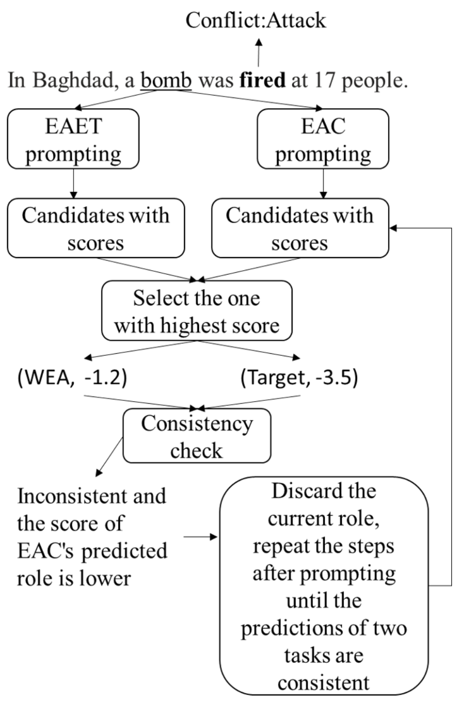
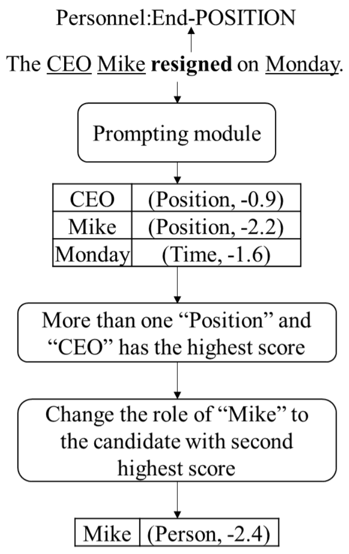

Conflict:Attack
In Baghdad, a bomb was fired at 17 people.
EAET prompting
EAC prompting
Candidates with scores
Candidates with scores
Select the one with highest score
(WEA, -1.2)
(Target, -3.5)
Consistency check
Inconsistent and the score of EAC's predicted role is lower
Discard the current role, repeat the steps after prompting until the predictions of two tasks are consistent

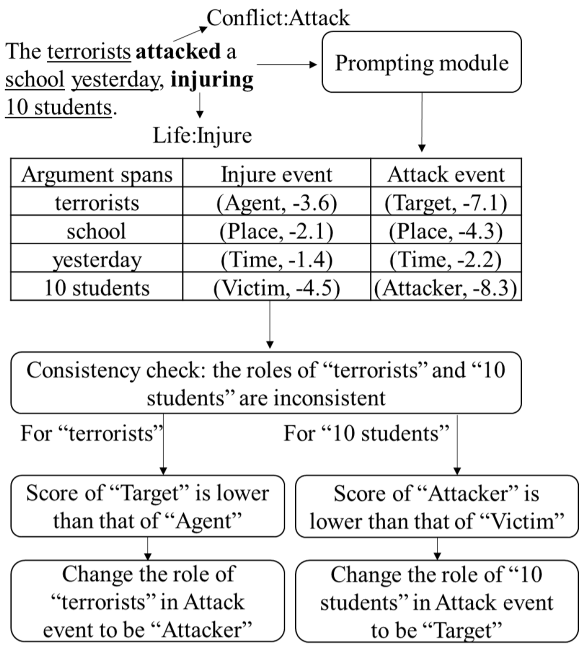
Figure 3: An Example of cross-task constraint. The text in **bold face** is the trigger, underlined text is target argument span, and a tuple denotes a predicted label with its prompting score (e.g., "(Target, -3.5)" denotes the predicted label "Target" with its prompting score"-3.5"). Similar notations are adopted in all remaining figures.

Our constraint modeling method can be easily generalized to other datasets/ontologies by simply using the knowledge about corresponding cross-task, cross-argument, and cross-event relations to design new constraints. The design processes are not costly as we could easily find such knowledge



Personnel:End-POSITION
The CEO Mike resigned on Monday.
Prompting module
CEO (Position, -0.9)
Mike (Position, -2.2)
Monday (Time, -1.6)
More than one "Position" and "CEO" has the highest score
Change the role of "Mike" to the candidate with second highest score
Mike (Person, -2.4)

Figure 4: An Example of cross-argument constraint.



Conflict:Attack
The terrorists attacked a school yesterday, injuring 10 students.
Life:Injure
Prompting module
Argument spans | Injure event | Attack event
terrorists | (Agent, -3.6) | (Target, -7.1)
school | (Place, -2.1) | (Place, -4.3)
yesterday | (Time, -1.4) | (Time, -2.2)
10 students | (Victim, -4.5) | (Attacker, -8.3)
Consistency check: the roles of "terrorists" and "10 students" are inconsistent
For "terrorists"
For "10 students"
Score of "Target" is lower than that of "Agent"
Score of "Attacker" is lower than that of "Victim"
Change the role of "terrorists" in Attack event to be "Attacker"
Change the role of "10 students" in Attack event to be "Target"

Figure 5: An Example of cross-event constraint.

from the guidelines of the target dataset.

3 Experiments

We first present the experimental settings, baselines used for comparison, and some implementation details. Next, we show and analyze the experiment results. Then we present a detailed analysis of the prompting module and global constraints regularization module. Finally, we conduct an error analysis.

3.1 Settings

We use ACE (2005-E^{+})[6] (Doddington et al., 2004; Lin et al., 2020) and ERE(-EN) (Song et al., 2015) as datasets. In total, ACE has 33 event types and 22 roles, whereas ERE has 38 event types and 21 roles. We pre-process all events to keep only the event subtypes whenever applicable, as done in (Lin et al., 2020). Following the pre-processing in (Zhang et al., 2021), for each dataset, we merge all splits into one test set since our approach is zero-shot. When argument spans are not given, we pipeline our model with an argument identification module adapted from (Lyu et al., 2021). Specifically, we replace the QA model in (Lyu et al., 2021) with a more powerful PTLM with a span classification head on top, and the whole model has been fined-tuned for extractive QA tasks. Then for a passage, we prompt each role using the new QA model as in (Lyu et al., 2021). We collect the prompt results for all roles (ignoring the "None" result) as candidate spans for the passage. We use the F1 score for evaluation following (Ji and Grishman, 2008), where argument spans are evaluated on the head level when not given. Regarding PTLMs, We use GPT-J (6B) (Wang and Komatsuzaki, 2021) instances from Huggingface (Wolf et al., 2020), where an instance for causal language modeling is used for prompting, and an instance for QA is used for argument identification. In all the following sections except Section 3.2, we conduct experiments on ACE, assuming that argument spans are given.

3.2 Main Results

We report the main results comparing our models with three previous powerful zero-shot models (Liu et al., 2020; Lyu et al., 2021; Zhang et al., 2021). Moreover, we also report the results of a SOTA supervised model (Hsu et al., 2022). We obtain the results of all compared methods from our own experiments to ensure a fair comparison on the same datasets and same settings. From Table 1, we have the following observations:

- Our model achieves superior performance on both datasets under both settings compared with all zero-shot baselines. Specifically, our model surpasses the best zero-shot baselines (Zhang et al., 2021) by 12.5% and 10.9% on ACE and ERE, respectively. Without argument spans, our model outperforms the respective best zero-shot baselines (Lyu et al.,

[6]https://www.ldc.upenn.edu/collaborations/past-projects/ace

Model	ACE		ERE	
	argument span given	argument span not given	argument span given	argument span not given
(Hsu et al., 2022) (supervised)	79.3	71.8	79.8	72.5
(Liu et al., 2020)	46.1	24.2	40.9	22.8
(Lyu et al., 2021)	47.8	26.9	44.5	26.3
(Zhang et al., 2021)	53.6	23.5	51.9	20.2
Ours	**66.1**	**31.2**	**62.8**	**29.6**

Table 1: Performance of supervised model, zero-shot baselines, and our model. The best scores among the ones of zero-shot methods are in bold font.

2021) by 4.3% and 3.3% on ACE and ERE, respectively, which is also a noticeable gap. Such large performance improvements can be attributed to the following: (1) the prefix prompt guides the PTLM to effectively capture input's event-related perspective and trigger; (2) the cloze prompt leverages linguistic and commonsense knowledge stored in PTLM to improve its contextual understanding of event arguments; (3) the global constraints regularization incorporate global information and domain knowledge in inference. In Section 3.3, we compare the effects of using different PTLMs like BERT in the prompting module, and the results show that our model consistently outperforms previous zero-shot models, as shown in Table 1 and Figure 6.

- Compared with the supervised SOTA model (Hsu et al., 2022), there is still a significant gap between our model's performance and that it. Specifically, (Hsu et al., 2022) outperforms our model by 13.2% and 17.0% on ACE and ERE, respectively. When argument spans are not provided, (Hsu et al., 2022) outruns our model by 40.6% and 42.9% on ACE and ERE, respectively. We can see that the advantage of supervised SOTA over our zero-shot method is much more distinct when argument spans are not given in advance. This is probably because our zero-shot argument identification module described in Section 3.1 is not powerful enough, which causes severe error propagation to our EAC model.

3.3 Analysis of Prompting Module

We conduct experiments to examine the effects of different configurations of prefix prompt templates. Specifically, we compare our model's complete prefix prompt with the following configurations: (1) removing event type information from the prefix; (2) removing trigger information from the prefix; (3) removing the whole prefix. For instance, suppose the passage is "In Baghdad, a bomb was fired at 17 people." mentioned in Section 2.2, the prefix in configuration (1) would be **"This event's occurrence is most clearly expressed by 'fired'."**, the prefix in configuration (2) would be **"This is a Attack event."**, and in configuration (3) there would be no prefix. The corresponding results are shown in Table 2, where we have the following observations. First, removing either event type or trigger from the prefix prompt will cause a performance drop, which indicates that both kinds of information have contributions to the prompting process. Second, event type plays a more significant role than trigger does in prefix prompt, and the joint effect of them is greater than the sum of their respective effects.

Configurations	F1	Δ
complete prefix prompt	66.1	-
w/o event type	64.4	-1.7
w/o trigger	64.9	-1.2
w/o prefix prompt	62.8	-3.3

Table 2: Results of using different configurations of prefix prompt.

In addition, we examine the effects of using different PTLMs in the prompting module. We compare the following PTLMs with GPT-J (6B): BERT (large, uncased) (Devlin et al., 2019), RoBERTa (large) (Liu et al., 2019), BART (large) (Lewis et al., 2020), GPT-2 (xl) (Radford et al., 2019), T5 (11B) (Raffel et al., 2020). The results are shown in Figure 6, where we have the following observations. First, the instance using GPT-J has the best performance, surpassing other instances by 4.2% to 7.9%. This shows that GPT-J has a better ability to understand events and their associated arguments compared to other PTLMs. Second, as PTLMs are listed in ascending order based on their numbers of

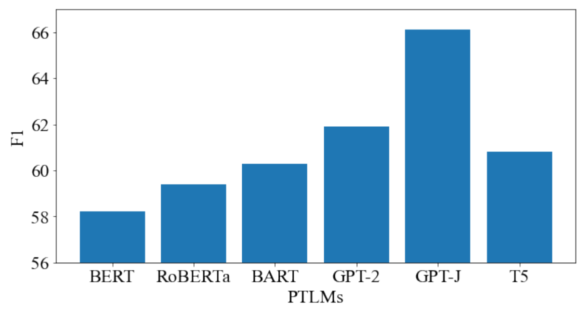


Figure 6: Comparison between the performance of using different PTLMs in prompting module.

parameters, we can see that for the first five models, the performance increases as the sizes of PTLMs become larger, which is consistent with the widely accepted notion that the larger model has a better capability of solving language tasks. However, the instance using the largest PTLM, T5 (11B), has a worse performance than GPT-2 and GPT-J. This is probably because autoregressive language modeling is more suitable for capturing information related to event arguments than mask language modeling is.

3.4 Analysis of Global Constraints Regularization Module

We conduct experiments to study the individual effect of each global constraint on the overall performance. The results are shown in Table 3, where we have the following observations. First, every

Model	F1	Δ
Full model	66.1	-
w/o cross-task constraint	60.5	-5.6
w/o cross-argument constraint	64.8	-1.3
w/o cross-event constraint	63.6	-2.5

Table 3: Results of using different configurations of global constraints.

global constraint used by our model is beneficial to overall performance, which demonstrates that exploiting the domain knowledge about cross-task, cross-argument, and cross-event relations indeed provides our model with global understanding of event arguments. Second, the contribution of cross-task constraint is the most significant, which suggests that the global insights from the entity typing tasks are more effective in improving our model's reasoning ability about event arguments. Third, the cross-argument constraint is less effective than the

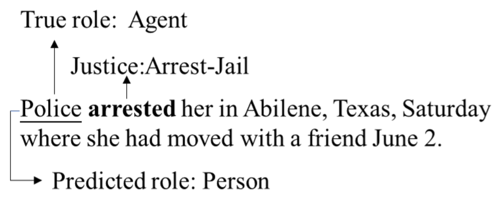

True role: Agent
Justice:Arrest-Jail
Police **arrested** her in Abilene, Texas, Saturday where she had moved with a friend June 2.
Predicted role: Person

Figure 7: An Example of the wrong prediction caused by too general argument roles. The text in **bold face** denotes trigger and the underlined text denotes target argument span.

other constraints, which shows that the global insights provided by the cross-argument constraint is less informative than those provided by the other constraints.

Apart from the three global constraints described above, we have designed another 11 global constraints, which rely on cross-argument or cross-event relations. We add each of them into our model to check their respective effects on the overall performance. The results of three of them are in Table 4, whereas the results of all of them are in Section B. From the results, we can find that each of these constraints either brings minor improvement or even has a negative influence on the overall performance. Hence, we do not incorporate these constraints in our model to maintain our model's efficiency and effectiveness.

3.5 Error Analysis

We manually checked 100 wrong predictions of our model and found that most of the errors are caused by too general roles of some event types. Specifically, some roles' linguistic meanings are so general that a model, not knowing their detailed event-type-dependent semantics, tends to assign them to some arguments which should have been assigned other roles. An example is shown in Figure 7. The example describes a Justice:Arrest-Jail event, which is associated with the following roles: "Person," "Agent," "Crime," "Time," and "Place." "Person" refers to the person who is jailed or arrested, whereas "Agent" refers to the jailer or the arresting agent. In the example, the argument span's true role should be "Agent" according to the detailed event-type-dependent semantics of "Person" and "Agent." However, our approach is zero-shot and directly models all role labels as natural language words, without incorporating the detailed event-type-dependent semantics of those roles, which are too general (e.g., "Person"). Therefore, our model assigns "Person" to "Police" since it is reasonable

Global constraint	Effect on overall performance
There is at most one Time-Arg in each event.	0.4
A TRANSPORT event has at most one ORIGIN argument.	-0.1
If an Arrest-Jail event and a Charge-Indict event share arguments, Arrest-Jail.Person is the same as Charge-Indict.Defendant, they share the same Crime argument.	0.3

Table 4: Results of three other global constraints. Results of all other global constraints are in Section B

from the perspectives of linguistic and commonsense knowledge, and "Person" is much more common than "Agent" in the pre-training corpus of the PTLM in the prompting module, which makes it have much higher likelihood in the language modeling process. Incorporating event-type-dependent semantics of the roles which are too general into our model is left as future work.

4 Related Work

In this section, we introduce related works about constraint modeling, event extractions, and prompt-based Information Extraction (IE).

4.1 Constraint Modeling

Constraint modeling, as an important technique in machine learning and NLP, aims to improve a model's performance by incorporating domain knowledge as constraints (Ganchev et al., 2010; Chang et al., 2012, 2013; Deutsch et al., 2019; Chang et al., 2008, 2010; Graça et al., 2010). One of the most significant advantages of constrained modeling is that it enables a model to capture the expressive and complex dependency structure in structured prediction problems like EAC (Chang et al., 2012). Especially in zero-shot scenarios, constrained modeling can provide useful indirect supervision to a model, which further boosts performance (Ganchev et al., 2010). Some previous works have adopted constraints based on event-related domain knowledge to classify event arguments (Lin et al., 2020; Zhang et al., 2021). However, their constraints either require labor-intensive annotations (Lin et al., 2020) or consider limited global information (e.g., cross-event relations) (Zhang et al., 2021). In this paper, our model uses global constraints to regularize prediction by incorporating global insights from cross-task, cross-argument, and cross-event relations.

4.2 Event Extraction

Event extraction is a fundamental information extraction task (Sundheim, 1992; Grishman and Sundheim, 1996; Riloff, 1996; Grishman et al., 2005; Chen et al., 2021; Du and Cardie, 2020; Liu et al., 2020), which can be further divided into four subtasks: trigger identification, trigger classification, argument identification, and argument classification. Traditional efforts mostly focus on the supervised setting (Ji and Grishman, 2008; Liao and Grishman, 2010; Liu et al., 2016; Chen et al., 2015; Nguyen et al., 2016; Liu et al., 2018; Zhang et al., 2019; Wadden et al., 2019; Lin et al., 2020). However, these works could suffer from the huge burden of human annotation. In this work, we focus on the argument classification task and propose a model using prompting and global constraints, without annotation and task-specific training.

4.3 Prompt-based IE

With the fast development of large PTLMs like T5 (Raffel et al., 2020), GPT-3 (Brown et al., 2020), and Pathway Language models (Chowdhery et al., 2022), the prompt-based method has been an efficient tool of applying those giant models into downstream NLP tasks (Liu et al., 2021). IE is not an exception. People have been using leverage prompts and giant models to solve IE tasks like named entity recognition (Cui et al., 2021), semantic parsing (Shin et al., 2021), and relations extraction (Chen et al., 2022; Han et al., 2021) in a zero-shot or few-shot way. However, previous prompting methods for IE need a tedious prompt design for every new type of events and arguments. In contrast, our model's prompt templates can be adapted to all possible types of events and arguments in a fully automatic way.

5 Conclusion

We propose a zero-shot EAC model using global constraints with prompting. Compared with previous works, our model does not require any annotation or manual prompt design, and our constraint modeling method can be easily adapted to any other datasets. Hence, our model can be easily generalized to any open-world event ontologies. Experiments on two standard event extraction datasets demonstrate our model's effectiveness.

6 Limitations

Our work has the following limitations. One limitation is that our model is not aware of the detailed event-type-dependent semantics of those roles which are too general, as discussed in Section 3.5. In the future, we will work on enabling our model to capture the event-type-dependent semantics of the roles which are too general. Another limitation is that our model's performance is still unsatisfactory compared with SOTA supervised model when argument spans are not given, as discussed in Section 3.2. In the future, we will work on designing a more powerful zero-shot event argument identification module for our model, so that we can obtain satisfactory zero-shot EAC performance even when argument spans are not given.

7 Acknowledgement

The authors of this paper were supported by the NSFC Fund (U20B2053) from the NSFC of China, the RIF (R6020-19 and R6021-20) and the GRF (16211520 and 16205322) from RGC of Hong Kong, the MHKJFS (MHP/001/19) from ITC of Hong Kong and the National Key R&D Program of China (2019YFE0198200) with special thanks to HKMAAC and CUSBLT, and the Jiangsu Province Science and Technology Collaboration Fund (BZ2021065). We also thank the support from the UGC Research Matching Grants (RMGS20EG01-D, RMGS20CR11, RMGS20CR12, RMGS20EG19, RMGS20EG21, RMGS23CR05, RMGS23EG08).

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Kai-Wei Chang, Rajhans Samdani, and Dan Roth. 2013. A constrained latent variable model for coreference resolution. In *EMNLP*, pages 601–612. ACL.

Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Vivek Srikumar. 2010. Discriminative Learning over Constrained Latent Representations. In *NAACL*. ACL.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2012. Structured learning with constrained conditional models. *Machine learning*, 88(3):399–431.

Ming-Wei Chang, Lev-Arie Ratinov, Nicholas Rizzolo, and Dan Roth. 2008. Learning and inference with constraints. In *AAAI*, pages 1513–1518. AAAI Press.

Muhao Chen, Hongming Zhang, Qiang Ning, Manling Li, Heng Ji, Kathleen McKeown, and Dan Roth. 2021. Event-centric natural language processing. In *ACL Tutorial*, pages 6–14.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL*, pages 167–176. ACL.

Yulong Chen, Yang Liu, Li Dong, Shuohang Wang, Chenguang Zhu, Michael Zeng, and Yue Zhang. 2022. Adaprompt: Adaptive model training for prompt-based NLP. *CoRR*, abs/2202.04824.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings ofACL/IJCNLP*, pages 1835–1845. ACL.

Hongliang Dai, Yangqiu Song, and Haixun Wang. 2021. Ultra-fine entity typing with weak supervision from a masked language model. In *ACL/IJCNLP*, pages 1790–1799. ACL.

Daniel Deutsch, Shyam Upadhyay, and Dan Roth. 2019. A general-purpose algorithm for constrained sequential inference. In *CoNLL*, pages 482–492. ACL.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186. ACL.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In *IJCAI*, pages 2327–2333. AAAI Press.

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 *LREC*. ELRA.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 671–683. Association for Computational Linguistics.

Kuzman Ganchev, João GraÃ, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(67):2001–2049.

João Graça, Kuzman Ganchev, and Ben Taskar. 2010. Learning tractable word alignment models with complex constraints. *Comput. Linguistics*, 36(3):481–504.

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference- 6: A brief history. In *COLING*, pages 466–471.

Ralph Grishman, David Westbrook, and Adam Meyers. 2005. Nyu's english ace 2005 system description. *ACE*, 5.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. PTR: prompt tuning with rules for text classification. *CoRR*, abs/2105.11259.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. Degree: A data-efficient generative event extraction model. In *NAACL*. ACL.

Kuan-Hao Huang, I-Hung Hsu, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. Multilingual generative language models for zero-shot cross-lingual event argument extraction. In *ACL*, pages 4633–4646. ACL.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare R. Voss. 2018. Zero-shot transfer learning for event extraction. In *ACL*, pages 2160–2170. ACL.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *ACL*, pages 254–262. ACL.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880. ACL.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *ACL*, pages 789–797. ACL.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *ACL*, pages 7999–8009. ACL.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *EMNLP*, pages 1641–1651. ACL.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016. Leveraging framenet to improve automatic event detection. In *ACL*. ACL.

Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022. Dynamic prefix-tuning for generative template-based event extraction. In *ACL*, pages 5216–5228. ACL.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *EMNLP*, pages 1247–1256.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *ACL/IJCNLP (Short Papers)*, pages 322–332. ACL.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: prompting argument interaction for event argument extraction. In *ACL*, pages 6759–6774. ACL.

Sneha Mehta, Huzefa Rangwala, and Naren Ramakrishnan. 2022. Improving zero-shot event extraction via sentence simplification. *CoRR*, abs/2204.02531.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *NAACL-HLT*, pages 300–309. ACL.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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.*, 21:140:1–140:67.

Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. In *AAAI*, pages 1044–1049. AAAI Press.

Richard Shin, Christopher H. Lin, Sam Thomson, Charles Chen, Subhro Roy, Emmanouil Antonios Platanios, Adam Pauls, Dan Klein, Jason Eisner, and Benjamin Van Durme. 2021. Constrained language models yield few-shot semantic parsers. In *EMNLP*, pages 7699–7715. ACL.

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 *The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation, EVENTS@HLP-NAACL*, pages 89–98. ACL.

Beth Sundheim. 1992. Overview of the fourth message understanding evaluation and conference. In *MUC*, pages 3–21.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *EMNLP-IJCNLP*, pages 5783–5788. ACL.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Sijia Wang, Mo Yu, Shiyu Chang, Lichao Sun, and Lifu Huang. 2022. Query and extract: Refining event extraction as type-oriented binary decoding. In *Findings of ACL*, pages 169–182. ACL.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP (Demos)*, pages 38–45. ACL.

Hongming Zhang, Haoyu Wang, and Dan Roth. 2021. Zero-shot label-aware event trigger and argument classification. In *Findings of ACL/IJCNLP*, pages 1331–1340. ACL.

Tongtao Zhang, Heng Ji, and Avirup Sil. 2019. Joint entity and event extraction with generative adversarial imitation learning. *Data Intell.*, 1(2):99–120.

Weizhong Zhao, Jinyong Zhang, Jincai Yang, Tingting He, Huifang Ma, and Zhixin Li. 2021. A novel joint biomedical event extraction framework via two-level modeling of documents. *Inf. Sci.*, 550:27–40.

A Comparison between Different Prefix Prompts

In this section, we conduct experiments on ACE-2005 dataset to compare the effectiveness of using different prefix prompts in our models. We compare the following prefix prompts with the one discussed in Section 2.2: (1) “**This is a [] event whose trigger is "[]".**”; (2) “**The event type is [], and its occurrence is most clearly expressed by "[]".**”; (3) “**The event type is [] and the trigger is "[]".**”. The results are shown in Table 5, where “Prefix (0)” refers to the prefix prompt discussed in Section 2.2, whereas “Prefix (1)” refers to the first prefix prompt described in this section, and so on. From the table we can see that the prefix prompt described in Section 2.2 is the most effective one, which might be due to the fact that the prefix prompt not only is based on the definitions of events and triggers (Grishman et al., 2005), but also has a natural and smooth expression.

Prefix Prompt	F1
Prefix (0)	**66.1**
Prefix (1)	65.2
Prefix (2)	65.6
Prefix (3)	63.0

Table 5: Performance of different prefix prompts.

B Results of all Other Global Constraints

In this section, we present the results of all other global constraints. The results are shown in Table 6.

Global constraint	Effect on overall performance
There is at most one Time-Arg in each event.	0.4
There is at most one Place-Arg in each event.	0.1
A TRANSPORT event has at most one Destination argument.	-0.2
A TRANSPORT event has at most one ORIGIN argument.	-0.1
A START-POSITION event has at most one Person argument.	0.2
A START-POSITION event has at most one Entity argument.	-0.1
A START-POSITION event has at most one Position argument.	0.1
A End-POSITION event has at most one Person argument.	-0.2
If a Start-Position event and an End-Position event share arguments, then Start-Position.Person is the same as End-Position.Person, and Start-Position.Entity is the same as End-Position.Entity, Start-Position.Position is the same as End-Position.Position.	0.1
If an Arrest-Jail event and a Charge-Indict event share arguments, Arrest-Jail.Person is the same as Charge-Indict.Defendant, they share the same Crime argument.	0.3
If a Die event and an Attack event share arguments, then Die.Place is the same as Attack.Place, Die.Victim is the same as Attack.Target, Die.Instrument is the same as Attack.Instrument, Die.Time is the same as Attack.Time, Die.Agent is the same as Attack.Attacker.	-0.2

Table 6: Other global constraints and corresponding effects on overall performance.