Hybrid Semantic Type Representation for Zero-Shot Event Extraction

Anonymous ACL submission

Abstract

Event extraction is a significant task in natural language processing. However, it is labor-intensive to get annotation when generalizing to new event types and ontologies. In this paper, we propose the HTR (Hybrid Type Representation) framework for zero-shot event extraction. We make a distinction of the abstraction level between events and roles, analyze role semantics, and propose a new representation approach, LRDB (label-related description-based), which is effective for both argument classification and collaboration with trigger extraction. We conduct extensive evaluation on ACE2005 dataset and achieve state-of-the-art.

1 Introduction

Event Extraction (EE) is a challenging task of information extraction, with the task of extracting event types and their elements (trigger words and the corresponding arguments) from a sentence. An example from the standard EE dataset ACE2005 shown in Figure 1 with two events, where "arrived" is the trigger for event Movement:Transport, "Kelly" (Artifact) "Seoul" (Destination), "Beijing" (Origin) and "Friday" (Time) are the corresponding arguments, while "brief" is the trigger for event Contact:Meet, "Kelly" (Entity) and Yoon (Entity) are the corresponding arguments. There are four subtasks: identify event triggers and classify them into predefined event types (trigger extraction), and identify their corresponding arguments and classify them into the corresponding predefined role types (argument extraction).

Most of the works solve the problem with supervised methods (Li et al., 2013; Chen et al., 2015; Yang and Mitchell, 2016; Nguyen et al., 2016; Liu et al., 2018; Yang et al., 2019; Huang et al., 2021; Ahmad et al., 2021), which is unable to do with unseen event types without labor-intensive annotations, thus is inflexible and limited. To achieve transfer from seen types to unseen types without any additional annotations, zero-shot learning methods have been explored on computer vision domain (Zhang and Saligrama, 2015; Ba et al., 2015; Changpinyo et al., 2016; Chen et al., 2018), and natural language processing (Ma et al., 2016; Bapna et al., 2017; Obeidat et al., 2019; Zhang et al., 2020). For event extraction, some recent works (Huang et al., 2018; Zhang et al., 2021; Lyu et al., 2021; Liu et al., 2020; Lai et al., 2021) explored on zero-shot methods. Huang et al. (2018) and Lai et al. (2021) followed the common zero-shot learning setting while Liu et al. (2020), Zhang et al. (2021) and Lyu et al. (2021) explored the setting without any training, which is followed in this paper. However, the performance of the system is still far from satisfaction.

We take a look in the ACE guideline, where describes the trigger as the word that most clearly indicates event's occurrence (mostly verbs) and arguments are the event participants or attributes (entities or values). [1] See the example in Figure 1, triggers "arrived" and "brief" are verbs, while the arguments "Kelly", "Yoon", "Seoul" and "Beijing" are entities, "Friday" is a value. Inspired by Zhang et al. (2021) meanwhile, we argue that the event type semantics can be expressed by the trigger word (e.g. "arrived" shows there is a transporting), while the entity role type semantics is dependent on the pattern in the sentence rather than any specific word (e.g. roles of "Kelly" and "Yoon" can be exchanged by swapping their location).[2] We express it as the different abstraction level of event mentions, i.e. event semantics is a low-level abstraction of trigger words while role semantics is a high-level abstraction of arguments with patterns. Inspired by this difference, we argue that semantics of event types and role types should be represented with

[1] https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf

[2] we focus on entities in this paper, the following roles mentioned are both entities in default.

Figure 1: An example from ACE2005, different events have different colors ("Kelly" participates in both two events), trigger are bold italic and arguments are underlined, and arrows point to their class types.

different methods based on their abstraction level.

With the low-level abstraction, the semantics of triggers is specific and informative in linking to event semantics. Trigger words representation is also an effective representation for event types, as shown in Zhang et al. (2021), and we name it as anchor word-based representation method. However, with the high-level abstraction, role semantics is more based on the pattern, for which anchor words are much useless. We analyze that a role semantics, i.e. the pattern, is based on the interaction with other roles (e.g. "Kelly" brief "Yoon") and its behaviour or function in the event (e.g. "Seoul" serves as the place). Basically, the pattern can be summarized as under what circumstances who does what action or has what function. We decompose the pattern into three information components: 1) scene, i.e. the circumstance, which is related to the event; 2) entity type, i.e. who or what, which can be shared among roles; 3) character, i.e. what action or what function, which is unique for each role. We focus on character information of role semantics for its uniqueness. Inspired by zero-shot works with label-based methods (Ma et al., 2016), which utilizes semantics of the type name, and description-based methods (Obeidat et al., 2019), which encodes semantics of type description text, we design to combine the type name with the description text on emphasis of character information to form an informative role type description text. We further observe that while some labels are good indicators of character (e.g. "Destination"), some are too general (e.g. "Person", "Entity") to contain appropriate character information. For these types, we design to replace their label with a more appropriate one (when it exists). With our label-related description text, we can encode role types representation to express the pattern in a degree.

Of role semantics, the character information is dependent on scene information (e.g. character of 'Destination' is dependent on transporting), and implicitly reflects some event semantics. We utilize it to improve the semantic similarity measure between triggers and event types.

In this paper, we consider the different abstraction-level of event and role types, and propose the HTR (Hybrid Type Representation) framework for zero-shot event extraction. We analyze role semantics, decompose it into three information components and argue the importance of character information for its uniqueness. Based on the analysis, we propose a new type representation approach LRDB (label-related description-based method) for role types representation, which can contribute to both argument classification and trigger extraction. In our framework, we adopt different representation methods for triggers and arguments, event and role types based on their different abstraction level. We identify triggers and arguments based on pretrained srl model, and map them to the most similar event types or role types respectively based on the semantic similarity among them. We show that with our framework, we can easily adapt to new types with some trigger anchor words and appropriate description text, without any additional annotations.

Our contributions are:

- We propose the HTR (Hybrid Type Representation) framework, which differentiates triggers and arguments, event types and role types representation based on their different abstraction level.
- We propose to focus on character information of role semantics, and utilize the information dependence to improve trigger-event similarity measure.
- We propose a new representation approach LRDB (label-related description-based) for role types, which is effective.

2 Related Work

Most of event extraction works are based on supervised methods, i.e. training and testing on the same event ontology set (Li et al., 2013; Chen et al., 2015; Yang and Mitchell, 2016; Nguyen et al., 2016; Liu et al., 2018; Yang et al., 2019; Huang et al., 2021; Ahmad et al., 2021). However, they can't adapt to the new types without additional annotations. Huang et al. (2018) firstly proposed a zero-shot framework for event extraction, but the method learns function on some seen types, and relies on the structural similarity between seen types and unseen when testing. Lai et al. (2021) also explored zero-shot event extraction with some seen types training data, but their setting of unseen role types is unrealistic. There are some zero-shot methods for event extraction without any training data, which is also the setting in this paper. Zhang et al. (2021) proposed to acquire event types and role types representation with the anchor word-based method, which is improper for role types. Liu et al. (2020) proposed a QA-based method using type names in query template, but it suffered from the meaning lack of the general type names, which contributes the most errors. The new representation approach LRDB proposed in this paper can handle this problem. Lyu et al. (2021) followed QA-based argument extraction but found their model is intrinsically weak on "no-answer" situations, which are common in reality.

3 Task Definition

Following Zhang et al. (2021), we denote $\mathcal{E}$ and $\mathcal{R}$ as the overall sets of predefined event trigger types and argument role types, respectively. Each predefined event type (e.g., "Movement:Transport") $E \in \mathcal{E}$ is associated with several role types $R \in \mathcal{R}_E$. Given a sentence $s = w_1 w_2 \dots w_{|s|}$, the task of zero-shot event trigger identification (TI) is to identity trigger words $t_1, \dots, t_n$ in the sentence while for the task of argument identification (AI), it is to identity argument words $a_1, \dots, a_m$ corresponding to the selected trigger word t in the sentence. The task of trigger classification (TC) is to classify the selected trigger word t from s to the event type $E \in \mathcal{E}$ while for the corresponding task of argument classification (AC), it is to classify the selected argument word a from s to the role type $R \in \mathcal{R}_E$.

4 Approach Overview

The whole framework can be divided into three stages: type representation preparation, identification and classification. The latter two modules are pipeline of event extraction, shown in Figure 2.

We first prepare event types representation and role types representation. With the data and method provided by Zhang et al. (2021), selected trigger anchor words and retrieved anchor sentences are used to encode event types representation with BERT. For role semantics, we analyze and decompose it into three information components: scene, entity type and character, and emphasize the character information with a label-related description-based method. We analyze and refine the inappropriate role type names when possible, and embed them into the description text that lays emphasis on the character information. We acquire role types representation by encoding the label-related description text with a sentence-level encoder defsent (Tsukagoshi et al., 2021).[3] See Section 5 for details.

For triggers and arguments representation, we encode them following (Zhang et al., 2021). With prepared event and role types representation, we can measure the similarity of each trigger and argument to predefined event types or role types, respectively. The role types representation encoded based on character information also introduces some scene information implicitly for their dependence. We utilize the implicit scene information of role types representation to improve the semantic similarity measure between triggers and event types.

In the identification module, given a sentence, we first identify candidate triggers and the corresponding candidate arguments with a BERT-based Verb+Nominal SRL (Semantic Role Labeling) model, then filter the triggers and arguments of concerns.[4] Then, triggers are filtered based on semantic similarity comparing with predefined event types, which is measured based on the cosine scores between triggers and event types representation, and between triggers and role types representation under the same event type. We filter arguments based on a selected subset of srl roles. The detail shown in Section 6.

In the classification module, we classify all triggers and the corresponding arguments output by the last stage. We map a trigger word to the most similar event type based on the same semantic similarity measured at the identification stage. For argument classification, we divide all of roles into specific and common groups, and use the predicted result by SRL for roles in common group. For roles in specific group, we map an argument to the most similar role type based on semantic similarity comparing with predefined role types corresponding to the classified event type at the trigger classification stage. The similarity between arguments and

[3] defsent encodes the sentence into the same semantic space shared with BERT, which allows for calculation between words and sentences.

[4] https://github.com/CogComp/SRL-English

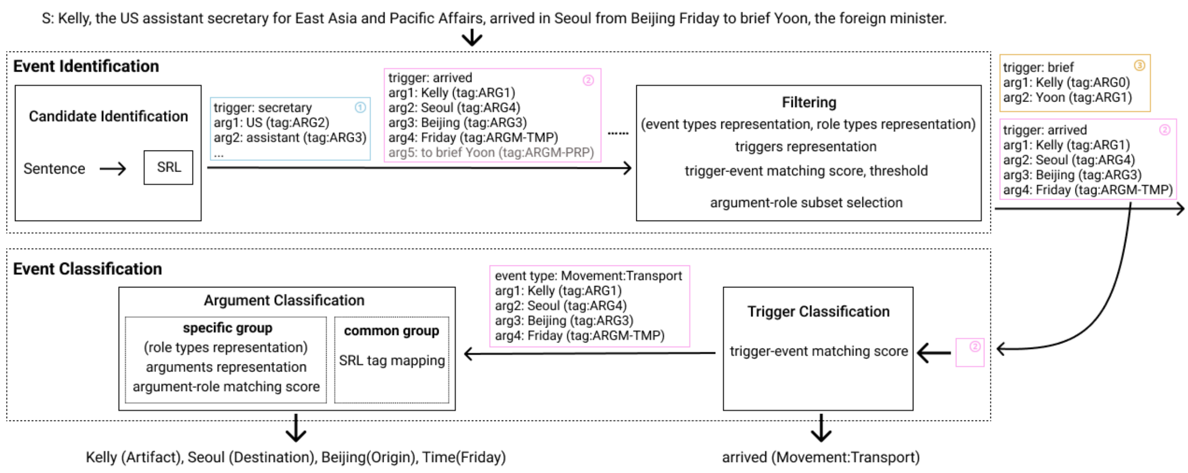

Figure 2: Event extraction pipeline in our framework with an example from ACE2005, we only show the event classification with one event type for simplicity.

role types is measured by cosine scores of their representation. The detail shown in Section 7.

5 Type Representation

We use different representation methods for event types and role types based on their different abstraction level. Specifically, we use anchor word-based method for event types representation, and label-related description-based method for role types representation.

5.1 Event type representation

With the low-level abstraction, the semantics of triggers is specific and informative in linking to event semantics. Following Zhang et al. (2021), we use the anchor word-based method, which selects some anchor trigger words for each event type, encode the contextualized representation of them in the retrieved anchor sentences, and cluster for event types representation.

5.2 Role type representation

With the high-level abstraction, we decompose the role semantics into three information components: scene, entity type and character, and focus on character information. To better utilize the semantics in label (type name) and description text, we design to acquire label-related description text for role types, and encode the text with defsent as role types representation.

Label-related description With a role type name, an initial role type description text and an event type description text, our task is to embed an appropriate type label (when there exists) into the role type description text, and simply modify the description text to emphasize the detailed character information. Some roles may have different character (according to the initial role type description or event type description), we detach the multi-character into multi description text for them, and encode each text a representation for the role type. See an example in Table 1.

The role types representation encoded based on character information also introduces some scene information implicitly for their dependence, which is utilized to improve the semantic similarity measure between triggers and event types in Section 6.

6 Identification

Following Zhang et al. (2021) and Lyu et al. (2021), we first identify candidate triggers and the corresponding candidate arguments based on a BERT-based Verb+Nominal SRL model, then further filter all predicates and arguments provided by SRL model for triggers and arguments of concerned.

Trigger filtering We first match each candidate triggers with concerned event types. The semantic similarity score s_{tE} between a trigger word t and an event type E is defined as the linear combination of comparing with event type representation and role types representation of the event type. Then, we filter the triggers above a similarity threshold.

$$s_E = cosine(emb_t, emb_E) \quad (1)$$

$$s_R = \frac{1}{n_{\mathcal{R}_E}} \sum_{r \in \mathcal{R}_E} cosine(emb_t, emb_r) \quad (2)$$

Event type description	A MEET Event occurs whenever two or more Entities come together at a single location and interact with one another face-to-face, include talks, summits, conferences, visits, ...
Role type name	Entity
Initial role type description	The agents who are meeting.
Label-included description	The *entities* or the agents who are meeting.
Label-refined description	The *participants* who are meeting.
Character-emphasis description	The *meeting* participants who *talk or visits, interacting face-to-face*.
Final text (detach multi-character)	The meeting participants who *talk*, interacting face-to-face. The meeting participants who *visit*, interacting face-to-face.

Table 1: An example shows acquisition of label-related description for the role "Entity" in event "Contact:Meet". Both the event type description and initial role type description are from ACE guideline.

$$s_{tE} = s_E + w_r * s_R \tag{3}$$

Argument filtering Similar to Huang et al. (2018), we manually select a subset of SRL roles of concerned. We build their mapping into event-related roles, as shown in Table 2. We filter the arguments whose SRL predicted roles are in this set.

Group	Role types	SRL roles
Common	Time Place	ARGM-TMP ARGM-LOC
Specific	Artifact, Destination, ...	ARG(0-8), ARGM-DIR

Table 2: Event-related SRL roles and their mapping.

7 Classification

Given the identified triggers and the corresponding arguments, we classify triggers to the event types and the corresponding arguments to the role types.

Trigger Classification We use the same score s_{te} acquired at the trigger filtering stage as the semantic similarity measure between the trigger and the event type. We classify a trigger to the event type with the highest similarity score.

$$E = \arg\max_{e} s_{te} \tag{4}$$

Argment Classification We divide all of roles into specific and common groups, and the roles in common group have specific tags in SRL, shown in Table 2. We use the common tags predicted by SRL model as the classification result, and only remain roles in specific group. The semantic similarity score s_{aR} between an argument and a role type is defined as cosine between argument representation and role type representation. And we classify a argument to the role type with the highest similarity score.

$$s_{aR} = cosine(emb_a, emb_R) \tag{5}$$

$$R = \arg\max_{r} s_{ar} \tag{6}$$

8 Experiment

Dataset, Setting and Evaluation We evaluate our methods on ACE2005 dataset, which has 33 event subtypes and 28 role types, including both entity and value roles. We use the same data split as in Zhang et al. (2021) and Lyu et al. (2021). Our method do not need any training, but need validating to make several design choices and select the hyper-parameters. Huang et al. (2018) trained in top-N most popular event types and tested on the least-23 frequent types, and Lyu et al. (2021) tested on test data of all 33 event types. To compare with them, we only use train and development set of top-10 event types for validating, remaining test set of top-10 event types and all data of least-23 event types to test.[5] We provide three evaluation setting: (A) Evaluation on the all data of least-23 event types (23all); (B) Evaluation on the test data of all 33 event types (33test); (C) Evaluation on the test data of top-10 event types and all data of least-23 types (merge). We evaluate argument spans on the head level following compared works (Huang et al., 2018; Zhang et al., 2021; Lyu et al., 2021). We report Hit@1, Hit@3 and Hit@5 for event classification task, and precision, recall and

[5] we don't compare with Liu et al. (2020) and Du and Cardie (2020) since our framework can't do argument identification alone.

f1 for event extraction pipeline the same as the compared works.

Implementation Details We adopt BERT-Large to encode triggers and arguments representation. For event types representation, we random choose 1000 sentences for each anchor word, use k-mean to cluster with k=2, repeating 10 times and select the maximum cluster with the lowest wce, then we merge all selected clusters of anchor words from an event type to form a complete cluster, using its centroid as the type representation. For role types representation, we use the annotation from ACE guideline as event types description and initial role types description, and use defsent-bert-large-uncased-mean to encode label-related description for types representation. We obtain the head of arguments with a simple heuristics-based head identifier based on the AllenNLP Dependency Parser following Lyu et al. (2021). [6] We weight 0.7 for the score s_R in the equation 3, and set the threshold as 0.87 for trigger filtering, tuning on the validating set. For argument extraction, we ignore all the reference and conference of roles detected by SRL.

8.1 Standard Evaluation

We consider two settings with the previous works: event classification and the overall event extraction pipeline.

8.1.1 Event Classification

This setting treats trigger classification and argument classification as two separate ranking problems with gold TI and AI. In Table3, we compare with the following methods:

- **WSDE** (Huang et al., 2018): WSD-Embedding method, The simplest baseline that uses pretrained word sense embeddings to encode type names as event types and role types representation, matching directly.
- **TL-D** (Huang et al., 2018): Structural similarity-based method, which uses the same event types and role types representation as WSD-Embedding, and learns structural similarity measure with data of top-10 event types to match.
- **AW** (Zhang et al., 2021): Anchor word-based representation method, which treats event types and role types in the same way, selects some anchor words for each type and encodes their contextualized representation in anchor sentences as type representation.
- **LRDB** (Zhang et al., 2021): Label-related description-based representation method, which encodes both event types and role types representation with label-related description.

From the results, in all setting, 1) label-related description-based method shows its advantage in encoding role semantics effective for argument classification, outperforming the anchor word-based method by considerable margins (8.3%-13.6%) in Hit@1; 2) Anchor word-based method shows its advantage in encoding event semantics effective for trigger classification with considerable margins (8.4%-15.4%) over LRDB method, which supports the different abstraction-level between trigger-event and argument-role as the analysis above-mentioned; 3) Our hybrid representation method shows its advantage by combining these two methods effectively and utilizing scene information of role types representation in collaboration with trigger-event semantic similarity measure, improving trigger classification with 1.1%-2.2%.

8.1.2 Event Extraction Pipeline

This setting evaluates the overall event extraction pipeline. In Table 4, we compare with the following systems:

- **TL-D** (Huang et al., 2018): AMR-based identification system.
- **TE/QA** (Lyu et al., 2021): SRL & Textual Entailment-based trigger extraction and QA-based argument extraction system.

From the results, we observe that 1) when f1 of both TI, TC and AC is large lower than TL-D Huang et al. (2018), our framework outperforms them at the last stage AC; 2) When our framework with lower performance in AI compared to Lyu et al. (2021), we also outperform them at the last stage AC. These observations show the effectiveness of our solution for argument classification. We also report our result evaluated in the merge (C) setting.

8.2 Ablation

We show the effect of each component of our method in this section.

[6] https://demo.allennlp.org/dependency-parsing

Setting	Method	Trigger Classification(%)			Argument Classification(%)		
		Hit@1	Hit@3	Hit@5	Hit@1	Hit@3	Hit@5
23all (A)	WSDE	1.7	13.0	22.8	2.4	2.8	2.8
	TL-D	33.5	51.4	68.3	14.7	26.5	27.7
	AW	82.1	89.1	93.0	52.6	88.9	98.5
	LRDB	73.7	**93.1**	95.6	**60.9**	**91.7**	**98.9**
	Hybrid	**83.3**	90.1	**96.0**			
33test (B)	AW	82.9	93.8	96.2	44.4	86.0	96.4
	LRDB	67.5	82.0	86.5	**58.0**	**89.0**	**98.0**
	Hybrid	**85.1**	**95.5**	**98.3**			
merge (C)	AW	82.1	89.9	93.7	49.9	87.7	97.9
	LRDB	71.6	89.9	93.0	**59.9**	**90.9**	**98.7**
	Hybrid	**83.2**	**91.1**	**96.4**			

Table 3: The Comparison of different methods of event classification in different setting on ACE2005

Setting	Method	TI(%)			TC(%)			AI(%)			AC(%)		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
23all (A)	TL-D	**85.7**	41.2	**55.6**	**75.5**	36.3	**49.1**	**28.2**	27.3	**27.8**	**16.1**	15.6	15.8
	HTR	32.9	**59.5**	42.3	29.2	**52.9**	37.6	18.6	**32.7**	23.8	14.9	**26.1**	**18.9**
33test (B)	TE/QA	34.7	**66.3**	45.5	31.7	**60.6**	41.7	20.2	**40.4**	**27.0**	12.6	**25.2**	16.8
	HTR	**51.8**	45.3	**48.3**	**48.2**	42.2	**45.0**	**29.3**	23.1	25.8	**20.5**	16.1	**18.1**
merge (C)	HTR	35.5	55.6	43.4	31.7	49.7	38.7	20.0	29.7	23.9	15.6	23.1	18.6

Table 4: The Comparison of different methods of event extraction pipeline in different setting on ACE2005.

8.2.1 Role type description

We show the effect of every step designed for role types description in Table 5 evaluating on argument classification.

	Hit@1	Hit@3	Hit@5
Initial	57.8	89.0	97.4
Label-included	57.3	91.1	98.0
Label-refined	59.4	90.3	98.2
Character	59.3	89.4	97.9
Detach	59.9	90.9	98.7

Table 5: Each step of role type description on argument classification in merge (C) setting

From the results, 1) simple annotation "Initial" is already better than the anchor word-based method (49.9%, in Table 3), which shows the advantage of description on the expression of role semantics; 2) The performance for "Initial", "Label-included" and "Label-refined" shows that simply embedding all type names indiscriminately into description text may hurt the role semantics while embedding with refined labels can contribute to it, indicating the importance of correct label; 3) The performance for the last three steps shows the importance of detaching multi-character into groups.

We further show the effect of correct labels and character-information description for role semantics in Table 6. We use random classification strategy as the baseline. From the results, 1) Compared with baseline, we can see from "Label-only" and "W/O-Label-Description" that both type names and description can contribute the role semantics;[7] 2) The comparison of "Label-only" and "Refined-Label" shows the importance of appropriate role type names; 3) The performance of "LR-Description" outperforms both refined label-only and description-only with large margins (12.1% and 14.1% respectively), showing the effectiveness of combining refined labels and description text.

8.2.2 Trigger-Event Sementic Similarity

We show the effect of role types representation on trigger-event semantic similarity measure in Table 7. We use the random trigger filtering and classification strategy as baseline. We can observe that role types representation does encode some event-related information useful for trigger extraction.

[7] W/O-Label-Description is acquired by replacing all labels in the text with the general label "entity"

	Hit@1	Hit@3	Hit@5
Baseline	23.6	53.9	70.2
Label-only	43.9	81.5	97.0
Refined-Label	45.8	83.1	97.4
W/O-Label-Description	47.8	83.9	96.9
LR-Description	59.9	90.9	98.7

Table 6: Effect of label and description for role semantics on argument classification in merge (C) setting.

	TI(%)			TI+TC(%)		
	P	R	F1	P	R	F1
Baseline	2.6	42.2	4.8	0.1	1.7	0.2
R-only	15.5	31.5	20.8	10.0	20.4	13.4
E-only	30.5	**56.6**	39.7	27.5	**51.0**	35.8
E+R	**35.5**	55.6	**43.3**	**31.7**	49.6	**38.7**

Table 7: Effect of role types representation on trigger-event semantic similarity measure, evaluating on trigger extraction in merge (C) setting.

8.2.3 Argument classification with SRL

We show the effect of SRL in argument classification of event extraction pipeline. To make it more clear, we only report the result of the final argument classification stage in Table 8.

	P	R	F1
all matching	13.7	20.4	16.4
SRL for common groups	**15.6**	**23.1**	**18.6**

Table 8: Effect of SRL in argument classification, evaluating in event extraction pipeline in merge (C) setting with AC reported.

References

Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. 2021. Gate: Graph attention transformer encoder for cross-lingual relation and event extraction. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*.

Lei Jimmy Ba, Kevin Swersky, Sanja Fidler, and Ruslan Salakhutdinov. 2015. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 4247–4255. IEEE Computer Society.

Ankur Bapna, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2017. Towards zero-shot frame semantic parsing for domain scaling. *ArXiv preprint*, abs/1707.02363.

Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. 2016. Synthesized classifiers for zero-shot learning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5327–5336. IEEE Computer Society.

Long Chen, Hanwang Zhang, Jun Xiao, Wei Liu, and Shih-Fu Chang. 2018. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1043–1052. IEEE Computer Society.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

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

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*, pages 5257–5269, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lifu Huang, Heng Ji, Kyunghyun Cho, 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 (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics.

Viet Lai, Minh Van Nguyen, Heidi Kaufman, and Thien Huu Nguyen. 2021. Event extraction from historical texts: A new dataset for black rebellions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2390–2400, Online. Association for Computational Linguistics.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online. Association for Computational Linguistics.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332, Online. Association for Computational Linguistics.

Yukun Ma, Erik Cambria, and Sa Gao. 2016. Label embedding for zero-shot fine-grained named entity typing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 171–180, Osaka, Japan. The COLING 2016 Organizing Committee.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics.

Rasha Obeidat, Xiaoli Fern, Hamed Shahbazi, and Prasad Tadepalli. 2019. Description-based zero-shot fine-grained entity typing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 807–814, Minneapolis, Minnesota. Association for Computational Linguistics.

Hayato Tsukagoshi, Ryohei Sasano, and Koichi Takeda. 2021. DefSent: Sentence embeddings using definition sentences. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 411–418, Online. Association for Computational Linguistics.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299, San Diego, California. Association for Computational Linguistics.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, Florence, Italy. Association for Computational Linguistics.

Hongming Zhang, Haoyu Wang, and Dan Roth. 2021. Zero-shot Label-aware Event Trigger and Argument Classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1331–1340, Online. Association for Computational Linguistics.

Tao Zhang, Congying Xia, Chun-Ta Lu, and Philip Yu. 2020. MZET: Memory augmented zero-shot fine-grained named entity typing. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 77–87, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ziming Zhang and Venkatesh Saligrama. 2015. Zero-shot learning via semantic similarity embedding. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 4166–4174. IEEE Computer Society.