Trigger-Argument based Explanation for Event Detection

Yong Guan[1], **Jiaoyan Chen**[2], **Freddy Lecue**[3], **Jeff Z. Pan**[4]*, **Juanzi Li**[1]*, **Ru Li**[5]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Department of Computer Science, The University of Manchester, UK
[3]INRIA, France
[4]School of Informatics, The University of Edinburgh, UK
[5]School of Computer and Information Technology, Shanxi University, Taiyuan, China
gy2022@mail.tsinghua.edu.cn, j.z.pan@ed.ac.uk, lijuanzi@tsinghua.edu.cn

Abstract

A critical task for constructing event knowledge graphs is event detection (ED), which aims to identify events of certain types in plain text. Neural models have achieved great success on ED, thus coming with a desire for higher interpretability. Existing works mainly exploit words or phrases of the input text to explain models' inner mechanisms. However, for ED, the event structure, comprising of an event trigger and a set of arguments, provides more enlightening clues to explain model behaviors. To this end, we propose a **T**rigger-**A**rgument based **E**xplanation method (TAE), which can utilize event structure knowledge to uncover a faithful interpretation for the existing ED models at neuron level. Specifically, we design *group*, *sparsity*, *support* mechanisms to construct the event structure from structuralization, compactness, and faithfulness perspectives. We evaluate our model on the large-scale MAVEN and the widely-used ACE 2005 datasets, and observe that TAE is able to reveal the process by which the model predicts. Experimental results also demonstrate that TAE can not only improve the interpretability on standard evaluation metrics, but also effectively facilitate the human understanding.

1 Introduction

Event Detection (ED) aims at identifying event triggers with specific event types, which is the first and fundamental step for extracting semantic and structural knowledge from plain text (Ahn, 2006; Nguyen and Grishman, 2015). For instance, event mention "*The train driver was beaten over the head by a thug.*" in Figure 1 comprises an event trigger "*beaten*" and a set of arguments such as "*the train driver*", "*the head*" and "*a thug*". An ideal ED system is expected to detect "*beaten*" as an event trigger of the type Bodily_harm. Recently, with the growth of open source annotated datasets (Walker et al., 2006; Wang et al., 2020) and the development of deep learning technologies, deep approaches have become popular for tackling the ED problem (Nguyen et al., 2016; Wang et al., 2021). Despite their great performance, they are still opaque for people to comprehend the inner mechanisms.

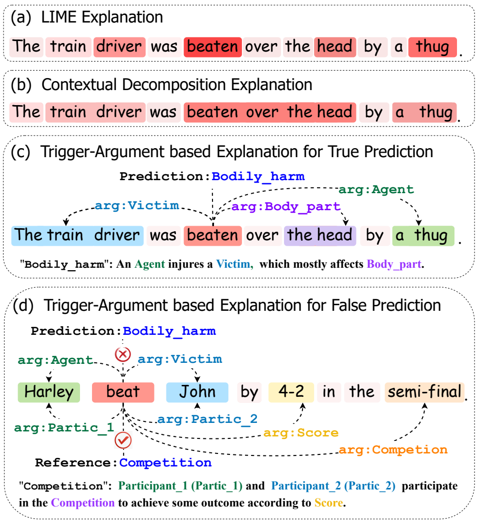

(a) LIME Explanation
The train driver was beaten over the head by a thug.
(b) Contextual Decomposition Explanation
The train driver was beaten over the head by a thug.
(c) Trigger-Argument based Explanation for True Prediction
Prediction:Bodily_harm
arg:Agent
arg:Victim
arg:Body_part
The train driver was beaten over the head by a thug.
"Bodily_harm": An Agent injures a Victim, which mostly affects Body_part.
(d) Trigger-Argument based Explanation for False Prediction
Prediction:Bodily_harm
arg:Agent
arg:Victim
Harley beat John by 4-2 in the semi-final.
arg:Partic_2
arg:Partic_1
arg:Score
arg:Competion
Reference:Competition
"Competition": Participant_1 (Partic_1) and Participant_2 (Partic_2) participate in the Competition to achieve some outcome according to Score.

Figure 1: Different explanations. For (a) and (b), features with deeper colors are considered more important by previous work. The usefulness of event triggers and arguments are illustrated in (c) and (d). "*arg*" refers to "*argument*". Bodily_harm and Competition are two event types in MAVEN.

Although there exist many works that focus on explaining the model behavior on natural language processing (NLP) problems, such as text classification (Lei et al., 2016), text matching (Jiang et al., 2021) and machine reading comprehension (Ju et al., 2021), very little progress has been made to interpret ED models. We identify two major limitations that prevent the existing explanation methods from being applied to ED models.

Neglecting event structured knowledge. Existing methods mainly focus on assessing the contributions of individual input unit (e.g., word or phrase) to generate explanations for neural networks (Li

*Corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2023, pages 5046–5058
July 9-14, 2023 ©2023 Association for Computational Linguistics

et al., 2016; Jiang et al., 2021). As shown in Figure 1(a) and (b), both explanations provide insights of which words (e.g., "*beaten*") or phrases (e.g., "*beaten over the head*") contribute to the prediction. However, neither of them is suitable to explain ED models as an event is represented as a structure comprising an event trigger and a set of arguments. Thus, the trigger-argument structures are more sensible clues to explain ED systems. In Figure 1(c), "*beaten*" is an ambiguous word that may evoke completely dissimilar events such as `Bodily_harm` and `Competition`. In this case, trigger word "*beaten*" and its arguments (e.g., "*the head*" and "*a thug*" which refer to `Body_part` and `Agent`) work together for the prediction `Bodily_harm`. Thus, how to take advantage of the event structure knowledge for ED model explanation is a non-trivial task.

Explanations cannot reflect the decision-making process. Models usually provide important features which are words or phrases selected from an input text as explanations, but they do not further elaborate the function of these features, i.e., why models produce the prediction according to these features. It poses challenges to interpret an explanation and connect it to model prediction. For example, in Figure 1(a) and (b), models may assign high relevance score to "*train driver*" or "*thug*", but it is still confused why these features can lead to the prediction `Bodily_harm`. In fact, "*train driver*" and "*thug*" serve as `Victim` and `Agent`, which compose together for the `Bodily_harm` event in which "*An* `Agent` *injures a* `Victim`" in Figure 1(c). Furthermore, Figure 1(d) provides an example that wrongly classifies `Competition` as `Bodily_harm`, because models take "*Harley*" and "*John*" as `Agent` and `Victim` rather than `Participant_1` and `Participant_2`. Thus, exploring explanations that can not only identify important features but also reveal how these features contribute to the prediction are urgently needed.

To address the aforementioned challenges, we propose **TAE**, a Trigger-Argument based Explanation method, to generate structure level explanations for ED models. Specifically, TAE focuses on utilizing neuron features of ED models to construct explanations based on trigger-argument knowledge. It has three core sub-modules: *Group Modeling* aims to divide neurons into different groups, where each group is regarded as an event structure, in such a way that each neuron corresponds to one argument and works together with other neurons that belong to the same event structure to explain the prediction of ED models; *Sparsity Modeling* aims to compact explanations by designing differentiable masking mechanisms to automatically filter out useless features generated by the group mechanism, and the intuition behind this module is that a good explanation should be short for understanding or reading (Miller, 2019); *Support Modeling* aims to ensure that the explanations generated by the group and sparsity mechanisms are faithful to the predictive model. Note we utilize FrameNet, a well-defined linguistic knowledge base by experts, to assist TAE identify event structures and help humans understand the decision-making process. The contributions of this paper are as follows:

- We propose a model-agnostic method, called TAE (**T**rigger-**A**rgument based **E**xplanation), to construct structure-level explanations for Event Detection (ED) systems. To the best of our knowledge, this is the first exploration to explain ED with structure knowledge.

- TAE adopts three strategies, namely, *Group Modeling*, *Sparsity Modeling* and *Support Modeling* to characterize the trigger-argument based explanations from structuralization, compactness, and faithfulness perspectives.

- We utilize FrameNet (Baker et al., 2006), a well-defined knowledge base, to help complete the event structure in MAVEN. The annotated data is released for further research[1].

- Experimental results on the large-scale MAVEN and widely-used ACE 2005 datasets show that TAE can generate more faithful and human-understandable explanations.

2 Related Work

In this section, we review the related works on *Event Detection* and *Interpretation Methods*.

Event Detection. Event detection is a key task for Event Knowledge Graph (Pan et al., 2017) construction. Traditional methods for ED have employed feature based techniques (Ji and Grishman, 2008; Li et al., 2013). These approaches mainly rely on elaborately designed features and NLP tools. Later, advanced deep learning methods have been applied for ED, such as convolutional neural networks (Chen et al., 2015), bidirectional

[1] https://github.com/neuroninterpretation/TAE

recurrent neural networks (Nguyen et al., 2016), which can take advantage of neural networks to learn features automatically. Since pre-trained language models (PLMs) are capable of capturing the meaning of words dynamically by considering their context, they have proven successful on a range of NLP tasks including ED (Tong et al., 2020; Wang et al., 2021, 2020). Although neural networks and PLMs bring incredible performance gains on ED task, they offer little transparency concerning the inner workings.

Interpretation Methods. There has been growing interests in producing explanations for deep learning systems in recent years, enabling humans to understand the intrinsic mechanism. In general, the explanations from these methods can typically be categorized as post-hoc explanations that aim to explain a trained model and reveal how model arrives at prediction (Lipton, 2016). Among them, gradient-based, attention-based and erasure-based methods are three typical methods.

Gradient-based methods are model-aware interpretation methods using gradients to measure feature importance (Shrikumar et al., 2017). Since a token index is not ordinal, methods simply sum up the relevance scores of each representation dimension. Because the score can have a negative or positive sign, the score may become zero even if it does contribute to prediction (Arras et al., 2017).

Attention-based methods attempt to use attention weights as feature importance scores (Vashishth et al., 2019). However, attention is argued to not be an optimal method to identify the attribution for an output as its validity is still controversial (Bastings and Filippova, 2020).

Erasure-based methods are widely-used approaches where a subset of features is considered irrelevant if it can be removed without affecting the model prediction (Feng et al., 2018). A straightforward approach is to erase each token by replacing it with a predefined value such as zero (Li et al., 2016). However, these erasure methods usually generate explanations by calculating the contribution of individual unit to the predictions, which are not suitable for ED as an event is often correctly identified with event structure.

In this paper, we attempt to generate explanations for ED models by considering semantic structured knowledge (Chen et al., 2018) entailed in the input at neuron level, which is complementary to the aforementioned approaches.

3 Preliminaries

3.1 Event Detection

An event refers to "*a specific occurrence involving one or more participants*" in automatic content extractions.To facilitate the understanding of the ED task, we introduce related terminologies as follows:

Event Trigger: the main word which most clearly expresses an event that happens.

Event Arguments: the entities that are involved in an event that serves as a participant.

Event Mention: a phrase or sentence within which an event is described.

For event mention "*The train driver was beaten over the head by a thug*", an event extractor is expected to identify an `Bodily_harm` event triggered by "*beaten*" and extract corresponding arguments with different roles such as "*the train driver*" (`Victim`) and "*a thug*" (`Agent`). In this paper, instead of explaining the overall standard event extraction models, we concentrate only on the ED task. That is, for this example, our goal is to explain why ED models can classify the event as `Bodily_harm` or not.

3.2 Problem Formulation

ED explanation aims to explain a trained ED model and reveal how the model arrives at the prediction. For an event mention $x = \{x_1, x_2, ..., x_i, ..., x_n\}$ with n words, a given pre-trained neural network (NN) f maps x to the corresponding label y_j, where $y_j \in \{y_1, y_2, ..., y_j, ..., y_m\}$ is corresponding event type which has unique trigger-arguments $F_x \in F$.

Assume that the NN model $f = g(h(x))$ can be decomposed into two-stages: (1) utilizes $h(\cdot)$ to map the input x to the intermediate layer $h(x) = \{h_1(x), h_2(x), ..., h_k(x), ..., h_N(x)\}$ with N neurons, such that $h(\cdot) \in \mathbb{R}^{(N \cdot d)}$, and $h_k(x)$ is the k-th neuron in $h(x)$; and (2) uses $g(\cdot)$ to map the intermediate layer $h(x)$ to the output $g(h(x))$, which is the probability of input x being predicted as label y_j by NN model, as shown in top part of Figure 2.

To better understand neurons, recent work attempts to identify the closest features to explain its behavior. The correlations between neuron and feature are obtained as follows:

$$Neu(h_k(x)) = \arg\max \rho(h_k(x), F) \tag{1}$$

where F are features of the input x, such as key words, POS, and trigger-arguments. $Neu(h_k(x))$ is the most related feature selected in x to represent $h_k(x)$. ρ is an arbitrary correlation calculation

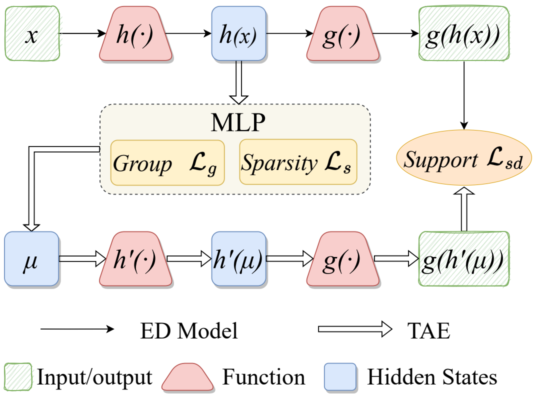


Figure 2: Architecture of TAE.

function, and we use IoU (intersection over union) in this paper. Following the existing work (Ghorbani et al., 2019; Mu and Andreas, 2020), we select the closest layer to the classifier that already learns more abstract information for prediction, to detect the neuron behavior[2].

4 Method

In this paper, we propose TAE, a trigger-argument based explanation method for event detection, which attempts to utilize event structure knowledge to explain model predictions at neuron level. The overview of our method is shown in Figure 2, which contains three modules: (1) The *Group* module captures structured knowledge of events; (2) The *Sparsity* module encourages models to select few but key features in the event structure; (3) The *Support* module is a fundamental module that guarantees explanations generated by *Group* and *Sparsity* consistent with the original prediction.

The loss function of an structured explanation for an event is obtained by an optimization problem:

$$\mathcal{L} = \arg\min \lambda_g \mathcal{L}_g + \lambda_s \mathcal{L}_s + \lambda_{sd} \mathcal{L}_{sd} \qquad (2)$$

where ($\mathcal{L}_g$, $\mathcal{L}_s$ and $\mathcal{L}_{sd}$) are from the group, sparsity and support modules, while λ_g, λ_s and λ_{sd} are hyper-parameters controlling weights of different losses.

4.1 Group Modeling

The Group module aims to divide neurons into different groups, and each group corresponds to a trigger-argument structure. Some existing works try to aggregate related features according to the distance information, such as encouraging the highly overlapping regions of the image (Varshneya et al., 2021), or gathering the neighbor words to enhance the current word (De Cao et al., 2020; Jiang et al., 2021). However, these methods might not work, as arguments of event types can be scattered in different positions and usually not adjacent to each other in input texts.

To solve this problem, we propose a group loss objective that constructs event structures by aggregating neurons corresponding to the related arguments. We first use the clustering algorithm k-means (Hartigan and Wong, 1979; Ghorbani et al., 2019) to automatically cluster neurons with the nearest mean into the same group.

$$G = \text{K-means}\left(\{h_k(x)\}\right) \qquad (3)$$

where $G \in \{G_1, G_2, ..., G_L\}$ is the group set and L is group number.

Then, for individual group G_l, we use the IoU to measure the contribution $\phi(h_i^l(x))$ of neuron $h_i^l(x)$ in the group.

$$\phi(h_i^l(x)) = \frac{2||h_i^l(x) - F_x||_1}{||h_i^l(x)||_1 + ||F_x||_1 + ||F_x - h_i^l(x)||_1} \qquad (4)$$

where F_x is the trigger-argument feature of input x, and $h_i^l(\cdot)$ is the i-th neuron in group G_l.

Finally, the group objective $\mathcal{L}_g$ is to minimize the intra-cluster sum of the distances from each neuron to the labeled feature in the input (Varshneya et al., 2021), given by the following equation:

$$\mathcal{L}_g = \sum_{l}^{L} \frac{1}{|G_l|} \sum_{i} \phi(h_i^l(x)) \qquad (5)$$

During train phase, for each batch, we extract the trigger-arguments on the whole batch while calculating the $\mathcal{L}_g$. It means that the neuron can learn the batch data features rather than individual features, which can enhance the generalization ability.

4.2 Sparsity Modeling

The Sparsity module aims to produce compact and human-friendly explanations. This is achieved by removing "dead neurons" (Mu and Andreas, 2020), which are useless for model prediction, while only keeping the key information to explain predictions.

To this end, following the existing work (De Cao et al., 2020), we use the differentiable masking mechanism to filter out the useless neuron features. Specially, for each extracted neuron, a classifier

[2] For a NN model, it has been shown that lower layers usually encode word and position information, and the higher layers can learn hierarchically-oriented information.

with a sigmoid activation function is added to determine whether the neuron should to be masked or not. During training phase, we directly use $L1$ norm (Jiang et al., 2021) to minimize the number of the neurons as follows:

$$\mathcal{L}_s = \min \sum_k \varphi(h_k(x)) \tag{6}$$

where $\varphi(\cdot)$ is the neuron classifier. The straightforward idea is to minimize the non-zero position.

4.3 Support Modeling

The support module aims to ensure the faithfulness of explanations generated by *Group* and *Sparsity*. A desirable interpretable event detection model should satisfy the intuition that a prediction is directly dependent on the selected features. For an ED model, we choose the neurons in $h(x)$ to generate explanations. Group and Sparsity are utilized to select neuron features μ containing structured and important information. Thus the goal of Support is to measure whether μ can depict the true profile of how the model works. Specifically, function $h'(\cdot)$ maps μ to the new hidden states $h'(\mu)$, and $g(\cdot)$ maps the new hidden states $h'(\mu)$ to the new output $g(h'(\mu))$, as shown in the bottom part of Figure 2.

We introduce an optimization objective to guarantee the support modeling. Different from the existing work matching the current prediction, we directly ask the reconstructed representation to meet the ground truth distribution[3].

$$\mathcal{L}_{sd} = \mathcal{P}(\hat{y}|g(h'(\mu), \theta)) \tag{7}$$

$$s.t. \quad KL(g(h(x)), g(h'(\mu))) \tag{8}$$

where $\hat{y}$ and θ are the ground truth labels and trainable parameters respectively. KL represents Kullback–Leibler divergence.

Note $h'(\cdot)$ can be any current popular network architectures, such as LSTM, Transformer and PLMs. In our setting, to maintain the interpretability, we use the simple linear projection and MLP (multilayer perceptron) to build the network, and the computation is much more efficient since we don't need to optimize the whole backbone (Yeh et al., 2020). In addition, in this way, it mainly focuses on learning the neuron behavior instead of sacrificing the performance of the pre-trained CNN models.

Methods	MAVEN			ACE 2005		
	P	R	F1	P	R	F1
LSTM	51.3	52.4	51.5	63.4	66.8	64.8
LSTM+CNN	53.5	54.2	52.3	65.6	67.2	64.9
BERT	52.6	63.5	57.7	69.9	72.2	70.5
DMBERT	53.1	65.2	58.6	71.9	**74.7**	71.4
DeBERTa	**58.7**	**65.6**	**60.8**	**73.7**	74.4	**72.1**

Table 1: Model performance on MAVEN and ACE. P and R refer to Precision and Recall respectively.

5 Experiment

5.1 Datasets

We evaluate our models on MAVEN (Wang et al., 2020) and ACE 2005 dataset (Walker et al., 2006).

MAVEN is a manually annotated dataset[4] for event detection task without annotated arguments, which contains 168 event types and 4,480 documents. The event types are manually derived from the frames defined in FrameNet (Baker et al., 1998; Guan et al., 2021b,a). To satisfy our needs, we utilize the automatic frame parser SEMAFOR (Das et al., 2014) to parse the MAVEN data. We select the data that have event type in MAVEN, and regard the corresponding frame elements (Guo et al., 2020) as the event arguments. Finally, we collect 12,649 event mentions, and randomly split them into train/dev/test sets with sizes of 8,469/2,000/2,000.

ACE 2005 is also a manually annotated dataset[5], which contains 8 event types, 35 argument roles and 599 English documents (Li et al., 2020). We further remove the data without arguments, and finally select 3,014 examples. Since the data size is relatively small, and cannot use it to learn a better NN model. So we directly utilize the models trained on MAVEN to test on ACE 2005, which can also verify the models' generalization ability.[6]

5.2 ED Models

Our TAE is a model-agnostic method to explain ED models. In this paper, we first select two typical NN models, namely LSTM (Hochreiter and Schmidhuber, 1997) which contains 2 layers with 300 hidden states, LSTM+CNN (Tan et al., 2015) which has 2 layers with 300 hidden states. Moreover, we also select three PLM-based models which

[3] Assume that we extract neurons from the pre-trained model, and the neuron exactly meets the current prediction. If we detect the trigger-argument information to be useful for model decision and remove the useless neurons, a reasonable explanation may meets or better than the current prediction.

[4] https://github.com/THU-KEG/MAVEN-dataset

[5] https://github.com/limanling/m2e2

[6] Because the event types on MAVEN and ACE 2005 are different, and can not directly test ACE data by the model which trained on MAVEN data. We use the mapping between ACE 2005 and MAVEN from (Wang et al., 2020) to get the final prediction.

Models	Explanations	ACE 2005				MAVEN			
		Support			Sparsity	Support			Sparsity
		AORC	AOPC	SUPP	SPAR	AORC	AOPC	SUPP	SPAR
LSTM	LEAVE-ONE-OUT	0.988	0.623	0.003	33.92	1.101	0.634	0.002	38.34
	BACKSELECT	1.103	0.614	0.005	36.72	1.124	0.682	0.005	40.17
	LIME	0.955	0.659	0.005	34.04	0.904	0.772	0.001	37.51
	DIFFMASK	0.913	0.717	0.019	**4.967**	0.903	0.891	0.013	**5.163**
	TAE(OURS)	**0.872**	**0.812**	**0.027**	5.220	**0.886**	**1.057**	**0.015**	5.313
LSTM+CNN	LEAVE-ONE-OUT	0.943	0.764	0.012	33.60	0.935	0.821	0.014	37.17
	BACKSELECT	0.981	0.776	0.018	33.89	0.954	0.846	0.011	38.65
	LIME	0.886	0.742	0.017	32.54	0.823	0.928	0.016	35.22
	DIFFMASK	0.801	0.799	0.021	4.841	0.778	1.116	0.027	**3.738**
	TAE(OURS)	**0.737**	**0.971**	**0.035**	**3.494**	**0.725**	**1.136**	**0.031**	4.244
BERT	LEAVE-ONE-OUT	0.841	0.893	0.026	26.06	0.915	1.076	0.040	29.29
	BACKSELECT	0.775	0.922	0.026	25.14	0.873	1.091	0.037	27.33
	LIME	0.714	0.954	0.035	23.44	0.809	1.139	0.041	26.07
	DIFFMASK	0.679	1.247	0.058	3.862	0.764	1.326	0.041	4.791
	TAE(OURS)	**0.535**	**1.453**	**0.072**	**2.471**	**0.693**	**1.557**	**0.048**	**2.926**
DMBERT	LEAVE-ONE-OUT	0.829	0.966	0.033	22.81	0.874	1.115	0.043	25.52
	BACKSELECT	0.767	0.979	0.029	21.47	0.822	1.156	0.044	23.82
	LIME	0.667	1.097	0.038	19.54	0.737	1.241	0.047	22.29
	DIFFMASK	0.517	1.207	0.048	2.733	0.662	1.464	0.046	3.429
	TAE(OURS)	**0.477**	**1.528**	**0.066**	**1.246**	**0.572**	**1.626**	**0.051**	**1.582**
DeBERTa	LEAVE-ONE-OUT	0.730	0.945	0.036	23.55	0.781	1.014	0.040	23.90
	BACKSELECT	0.700	0.971	0.033	20.15	0.743	1.002	0.043	23.35
	LIME	0.692	1.083	0.048	19.54	0.716	1.156	0.050	21.82
	DIFFMASK	0.616	1.221	0.054	1.938	0.719	1.155	0.058	2.437
	TAE(OURS)	**0.525**	**1.674**	**0.073**	**1.148**	**0.623**	**1.774**	**0.069**	**1.603**

Table 2: Support and Sparsity evaluation of different methods on ACE 2005 and MAVEN.

achieve promising performance on ED, including BERT (Devlin et al., 2019) which has 12 layers and 768 hidden states, DMBERT (Wang et al., 2019) which also applied on BERT-base version with 768 hidden states, and DeBERTa (He et al., 2021) which has 24 layers and 1,536 hidden states. Table 1 shows the results (P, R, F1) of different models on both datasets in our experiments, where DeBERTa outperforms the other four models with higher F1 scores.

5.3 Support Evaluation

We adopt three metrics to evaluate support degree (i.e., faithfulness): two metrics from prior explanation work including *area over reservation curve* (AORC) (DeYoung et al., 2020) and *area over the perturbation curve* (AOPC) (Nguyen, 2018), and a new defined evaluation metric called *support-score* (SUPP).

AORC calculates the distance between the original predicted logits and the masked ones by reserving top *k*% neuron features which are identified by trigger-arguments as follows:

$$\text{AORC} = \sum_{k=0}^{K} ||\mathcal{P}(\hat{y}|x) - \mathcal{P}_{(k)}{}''(\hat{y}|x)||_2 \quad (9)$$

where $\mathcal{P}_{(k)}{}''(\hat{y}|x)$ means the prediction which reserves the top *k*% neuron features. Under this metric, lower AORC scores are better.

AOPC score calculates the average change in prediction probability on the predicted class over all test data by deleting the top *r*% neuron features.

$$\text{AOPC} = \frac{1}{N}\sum_{i=1}^{N}\{\mathcal{P}(\hat{y}|x) - \mathcal{P}_{(r)}{}''(\hat{y}|x)\} \quad (10)$$

where $\mathcal{P}_{(r)}{}''(\hat{y}|x)$ is the prediction which remove the top *r*% neuron features. N denotes the number of examples. In our experiment, *r* is set to 20. Under this metric, the larger scores are better.

We propose SUPP score to verify whether the new prediction $g(h'(\mu))$ is positive to the original ones $g(h(x))$. Under this metric, the larger SUPP scores are better.

$$\text{SUPP} = \frac{1}{N}\sum\{g(h'(\mu) - g(h(x))\} \quad (11)$$

We compare TAE with four competitive baselines, namely LEAVE-ONE-OUT (Li et al., 2016), LIME (Ribeiro et al., 2016), BACKSELECT (Carter et al., 2019) and DIFFMASK (De Cao et al., 2020), utilizing AORC, AOPC and SUPP metrics.

Automatic support evaluation results are shown in Table 2, and we have the following three observations:

(1) TAE achieves better performance in most cases across all the three metrics on both datasets. For metric SUPP, all methods achieve positive results, indicating our method can identify important features and make a positive contribution to model predictions.

(2) Compare to LSTM- and CNN- based methods, BERT-based methods achieve significantly better performance. It is perhaps because BERT has

Methods	Attack	Arriving	Statement	Motion	Process_start	Creating	Death	Giving	Avg.
LSTM	0.0542	0.0628	0.0354	**0.0478**	0.0602	0.0686	**0.0537**	0.0489	0.0540
LSTM+Group	**0.0564**	**0.0675**	**0.0381**	0.0453	**0.0633**	**0.0705**	0.0526	**0.0545**	**0.0560**
LSTM+CNN	0.0564	0.0536	**0.0430**	0.0508	0.0627	**0.0439**	0.0556	0.0376	0.0505
LSTM+CNN+Group	**0.0615**	**0.0557**	0.0426	**0.0513**	**0.0725**	0.0437	**0.0598**	**0.0447**	**0.0540**
BERT	0.0746	0.0662	0.0577	0.0692	0.0701	0.0720	0.0510	0.0653	0.0658
BERT+Group	**0.0763**	**0.0683**	**0.0581**	**0.0758**	**0.0712**	**0.0739**	**0.0629**	**0.0772**	**0.0705**
DMBERT	0.0763	0.0651	0.0497	0.0833	0.0614	0.0720	0.0624	0.0668	0.0671
DMBERT+Group	**0.0789**	**0.0651**	**0.0586**	**0.0917**	**0.0721**	**0.0733**	**0.0640**	**0.0766**	**0.0725**
DeBERTa	0.0782	0.0654	0.0517	0.0862	0.0638	0.0695	0.0601	0.0676	0.0677
DeBERTa+Group	**0.0822**	**0.0655**	**0.0588**	**0.0917**	**0.0799**	**0.0710**	**0.0615**	**0.0747**	**0.0731**

Table 3: IoU scores for the 8 event types.

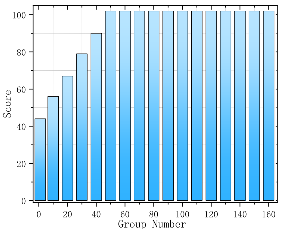


Figure 3: Interpretability score for TAE.

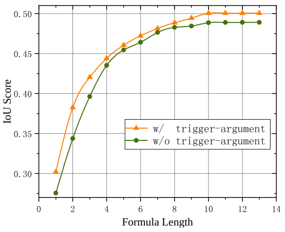


Figure 4: IoU scores w/ and w/o trigger-arguments.

already preserved a large amount of general knowledge by training on large-scale data.

(3) Compared with MAVEN, the results on ACE are equally remarkable. Overall, our model achieves very strong results on different types of data and methods, proving that it is a good model-agnostic approach.

5.4 Sparsity Evaluation

For evaluating the sparsity, we directly report the *sparsity score*, which obtained in Equation 6 just like the explanation work (Jiang et al., 2021), as the metric. In this criterion, the score means the degree of sparsity, and the lower scores are better. The intuition behind this criterion is that a good explanation should be short for understanding.

The results are reported in Table 2. We can see TAE achieves the lowest SPAR values in most cases for the automatic evaluation, for example, the SPAR of our TAE + DEBERTA on ACE and MAVEN are 1.603 and 1.148, while the SPAR of LEAVE-ONE-OUT are 23.90 and 23.55, which indicates that TAE can effectively discover the useless neurons.

5.5 Group Evaluation

In order to verify the effectiveness of the group mechanism, we use two metrics for explanations. First, following the previous explanation work (Bau et al., 2017; Varshneya et al., 2021), for each predefined group, we compute the number of unique trigger-argument in the group as the *interpretability score*. Second, for trigger-argument structure, we average the *IoU score* which is computed in Equation 1 to represent its explanation quality score like (Mu and Andreas, 2020).

Figure 3 shows the comparison of the interpretability score with different groups. With the group number increasing, the number of trigger-arguments detected by the model gradually increases, indicating grouping mechanism can improve the model interpretability. However, the number of trigger-arguments remains constant after the group number exceeds 50. A major reason is the uneven distribution of the data, which mainly concentrates on 20% of the event types. Note, the maximum group number is limited to event type number, such as for MAVEN, the maximum group number is 168.

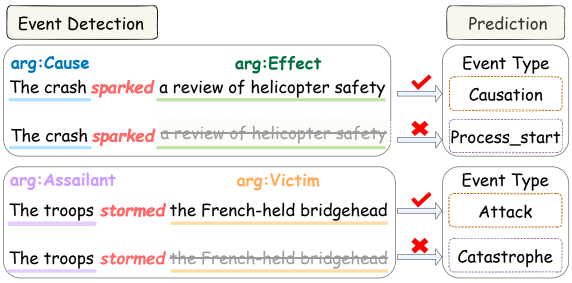
Table 3 shows the IoU score of 8 event types. In this setting, we separate test for each event type. The score increases with the group mechanism on most cases, which can further prove the effectiveness of the group mechanism.

5.6 Analysis on Trigger-Argument

To further verify the effectiveness of the trigger-arguments, we introduce features used in Mu and Andreas (2020) as comparison, such as POS (part-of-speech), most common words, and entity. They suggest that neuron cannot be regarded as a simple detector (Bau et al., 2017) but may express the meaning of multiple concepts. So they use composition operations such as OR, AND and NOT to expand the neuron behavior. We use the average IoU score of whole neurons on different formula lengths as one metric:

$$SL_i = \frac{1}{|h_x|} \sum_{j=0} \arg\max \mathrm{IoU}(h_j(x), F) \qquad (12)$$

where SL_i is the IoU score of the formula length i, and F is the feature set. $h_j(\cdot)$ denotes the j-



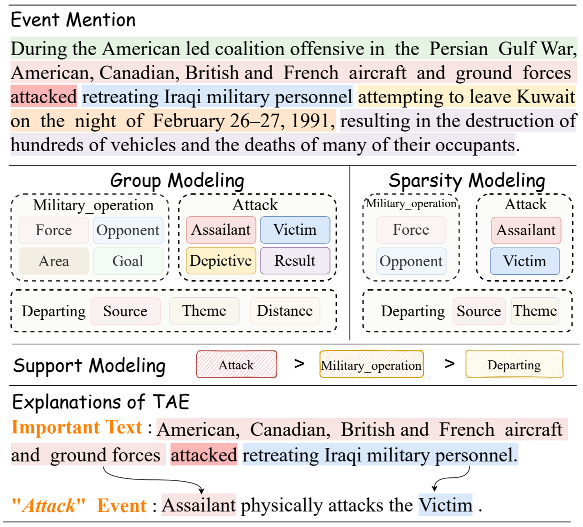
The crash sparked a review of helicopter safety ✓

The crash sparked ~~a review of helicopter safety~~ ✗

The troops stormed the French-held bridgehead ✓

The troops stormed ~~the French-held bridgehead~~ ✗

Figure 5: Examples of deleting important arguments.



During the American led coalition offensive in the Persian Gulf War, American, Canadian, British and French aircraft and ground forces attacked retreating Iraqi military personnel attempting to leave Kuwait on the night of February 26–27, 1991, resulting in the destruction of hundreds of vehicles and the deaths of many of their occupants.

Important Text : American, Canadian, British and French aircraft and ground forces attacked retreating Iraqi military personnel.

"*Attack*" Event : Assailant physically attacks the Victim .

Figure 6: An example of TAE explanation.

th neuron feature and $|h_x|$ is the neuron number. Under this metric, larger IoU scores are better.

Figure 4 shows the results with (w/) and without (w/o) trigger-arguments, and we obtain the following two findings: (1) with the help of trigger-argument, the IoU scores are larger than that w/o trigger-argument on each formula length. The results demonstrate that trigger-argument can help generate more faithfulness explanations compared to word level features. That's because each argument expresses complete meaning which may contain a semantic span rather than an individual word. (2) as the max formula length keeps getting larger, the IoU score keeps getting larger. When the formula length is greater than 10, the score is no longer changing, indicating the maximum representation capacity of neuron is 10 trigger-arguments for the model.

We further perform a qualitative analysis by deleting the arguments with high support scores. As shown in Figure 5, event mention "*The crash sparked a review of helicopter safety*" belongs to `Causation`. Explanation of our TAE model is that "*The crash*" and "*sparked a review of helicopter safety*" are two core arguments to form `Causation` that "*An Cause causes an Effect*". So when we delete argument `effect` ("*sparked a review of helicopter safety*"), ED model wrongly classifies the event as `Process_start`. The same applies to the second event `Attack`, when delete the `Victim`, ED models identify it as `Catastrophe`. The qualitative results indicate that our proposed TAE can capture trigger-argument structures that are important for model prediction.

5.7 Case Study

Figure 6 shows an example of TAE explanation. Given an event mention, 1) Group Modeling divides neurons into different event structure according to the arguments information, e.g., neurons are grouped into `Military_operation`, `Attack` and `Departing`; 2) Sparsity Modeling filters useless features such as `Depictive` and `Result` to compact the explanations; 3) Support Modeling selects features that are consist with the prediction, for example, `Attack` are more faithful comparing to `Military_operation` and `Departing`. From the above three procedures, we obtain the TAE explanation, which not only contains important features from the text but also reveals why they are important for the final prediction. For instance, "*American, Canadian, British and French aircraft and ground forces*" and "*retreating Iraqi military personnel*" respectively refer to `Assailant` and `Victim`, which work together to characterize the `Attack` event in which "`Assailant` *physically attacks the* `Victim`". In addition, with the help of trigger-argument information, the explanation is more helpful for human understanding.

6 Conclusion

In this paper, we propose a trigger-argument based explanation method, TAE, which exploits the event structure-level explanations for event detection (ED) task. TAE focuses on utilizing neuron features of ED models to generate explanations, along with three strategies, namely, group modeling, sparsity modeling, and support modeling. We conduct experiments on two ED datasets (i.e., MAVEN and ACE). The results show that TAE achieves better performance compared to the strong baselines on multiple public metrics. In addition, TAE also provides more faithful and human-understandable explanations.

There might be a few different future directions. Firstly, we might look into the idea of using explanations to further improve ED classification, as well as ED explanations in downstream applications. Secondly, we plan to explore ED under the multi-modal setting. Thirdly, event relation extraction is still challenging and deserves some further investigation. From the practical aspect of event knowledge graphs (EKGs), it is worth investigating high-quality yet efficient methods for constructing EKGs and making use of EKGs to predict future events (Lecue and Pan., 2013; Deng et al., 2020). Furthermore, it might be an idea to integrate commonsense knowledge (Speer et al., 2016; Romero et al., 2019; Malaviya et al., 2020) into event knowledge graphs.

Limitations

In this section, we discuss the limitations of TAE. First, as our method depends on event structure information which is obtained through automatic parser, if the parser is not good enough, then it will impact the performance. Second, since we focus on leveraging structural information, we restrict the experiments on text-based event explanation. Future work will explore multi-modal event detection explanations and evaluate models on other NLP tasks.

Acknowledgments

This work is supported by the Institute for Guo Qiang, Tsinghua University (2019GQB0003), the NSFC Youth Project (62006136), and the Chang Jiang Scholars Program (J2019032).

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8.

Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017. Explaining recurrent neural network predictions in sentiment analysis. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 159–168.

C. F. Baker, C. J. Fillmore, and J. B. Lowe. 2006. The berkeley framenet project.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 86–90.

Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3319–3327.

Brandon Carter, Jonas Mueller, Siddhartha Jain, and David Gifford. 2019. What made you do this? understanding black-box decisions with sufficient input subsets. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 567–576. PMLR.

Jiaoyan Chen, Freddy Lecue, Jeff Z. Pan, Ian Horrocks, and Huajun Chen. 2018. Knowledge-based Transfer Learning Explanation. In *Proc. of the International Conference on Principles of Knowledge Representation and Reasoning (KR2018)*, pages 349–358.

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*, pages 167–176.

Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56.

Nicola De Cao, Michael Sejr Schlichtkrull, Wilker Aziz, and Ivan Titov. 2020. How do decisions emerge across layers in neural models? interpretation with differentiable masking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 3243–3255.

Songgaojun Deng, Huzefa Rangwala, and Yue Ning. 2020. Dynamic knowledge graph based multi-event forecasting. In *Proc. of KDD*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings* *of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. 2019. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems*, volume 32.

Yong Guan, Shaoru Guo, Ru Li, Xiaoli Li, and Hongye Tan. 2021a. Frame semantic-enhanced sentence modeling for sentence-level extractive text summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4045–4052, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yong Guan, Shaoru Guo, Ru Li, Xiaoli Li, and Hu Zhang. 2021b. Integrating semantic scenario and word relations for abstractive sentence summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2522–2529, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shaoru Guo, Ru Li, Hongye Tan, Xiaoli Li, Yong Guan, Hongyan Zhao, and Yueping Zhang. 2020. A frame-based sentence representation for machine reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 891–896, Online. Association for Computational Linguistics.

John A Hartigan and Manchek A Wong. 1979. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262.

Zhongtao Jiang, Yuanzhe Zhang, Zhao Yang, Jun Zhao, and Kang Liu. 2021. Alignment rationale for natural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 5372–5387.

Yiming Ju, Yuanzhe Zhang, Zhixing Tian, Kang Liu, Xiaohuan Cao, Wenting Zhao, Jinlong Li, and Jun Zhao. 2021. Enhancing multiple-choice machine reading comprehension by punishing illogical interpretations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3641–3652.

Freddy Lecue and Jeff Z. Pan. 2013. Predicting Knowledge in An Ontology Stream. In *Proc. of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Manling Li, Alireza Zareian, Qi Zeng, Spencer Whitehead, Di Lu, Heng Ji, and Shih-Fu Chang. 2020. Cross-media structured common space for multimedia event extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2557–2568.

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*, pages 73–82.

Zachary Lipton. 2016. The mythos of model interpretability. *Communications of the ACM*, 61.

Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. 2020. Commonsense knowledge base completion with structural and semantic context. In *Proceedings of AAAI*.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38.

Jesse Mu and Jacob Andreas. 2020. Compositional explanations of neurons. In *Advances in Neural Information Processing Systems*, volume 33, pages 17153–17163.

Dong Nguyen. 2018. Comparing automatic and human evaluation of local explanations for text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1069–1078.

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.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with 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*, pages 365–371.

J. Z. Pan, G. Vetere, J.M. Gomez-Perez, and H. Wu, editors. 2017. *Exploiting Linked Data and Knowledge Graphs for Large Organisations*. Springer.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1135–1144.

Julien Romero, Simon Razniewski, Koninika Pal, Jeff Z. Pan, Archit Sakhadeo, and Gerhard Weikum. 2019. Commonsense Properties from Query Logs and Question Answering Forums. In *Proc. of 28th ACM International Conference on Information and Knowledge Management (CIKM 2019)*, pages 1411–1420.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153.

Robert Speer, Joshua Chin, and Catherine Havasi. 2016. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI Conference on Artificial Intelligence*.

Ming Tan, Bing Xiang, and Bowen Zhou. 2015. Lstm-based deep learning models for non-factoid answer selection. *CoRR*, abs/1511.04108.

Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020. Improving event detection via open-domain trigger knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5887–5897.

Saurabh Varshneya, Antoine Ledent, Robert A. Vandermeulen, Yunwen Lei, Matthias Enders, Damian Borth, and Marius Kloft. 2021. Learning interpretable concept groups in cnns. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 1061–1067.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019. Attention interpretability across nlp tasks. *arXiv preprint arXiv:1909.11218*.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium*.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019. Adversarial training for weakly supervised event detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 998–1008.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 1652–1671.

Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. 2021. CLEVE: Contrastive Pre-training for 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*, pages 6283–6297.

Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. 2020. On completeness-aware concept-based explanations in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 20554–20565.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations

☑ A2. Did you discuss any potential risks of your work?
Section 5

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 5.1

☑ B1. Did you cite the creators of artifacts you used?
Section 5.1

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 5.1

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 5

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 5

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 5

C ☑ Did you run computational experiments?

Section 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 5

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 5

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 5

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☒ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☒ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☒ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.