Saliency as Evidence: Event Detection with Trigger Saliency Attribution

Jian Liu, Yufeng Chen, Jinan Xu
Beijing Jiaotong University, School of Computer and Information Technology, China
{jianliu, chenyf, jaxu}@bjtu.edu.cn

Abstract

Event detection (ED) is a critical subtask of event extraction that seeks to identify event triggers of certain types in texts. Despite significant advances in ED, existing methods typically follow a "one model fits all types" approach, which sees no differences between event types and often results in a quite skewed performance. Finding the causes of skewed performance is crucial for the robustness of an ED model, but to date there has been little exploration of this problem. This research examines the issue in depth and presents a new concept termed trigger salience attribution, which can explicitly quantify the underlying patterns of events. On this foundation, we develop a new training mechanism for ED, which can distinguish between trigger-dependent and context-dependent types and achieve promising performance on two benchmarks. Finally, by highlighting many distinct characteristics of trigger-dependent and context-dependent types, our work may promote more research into this problem.

1 Introduction

Event detection (ED) is the first and a crucial step of event extraction, which aims to identify events of certain types in plain texts (Ahn, 2006; Nguyen and Grishman, 2015; Mitamura et al., 2017). Previous methods to ED typically adopt a "one model fits all types" approach, seeing no difference between event types and using a single model to address them all (Ji and Grishman, 2008; Li et al., 2013; Chen et al., 2015; Lin et al., 2020). However, such approaches produce quite skewed performance on different types. Tasking the ACE benchmark as an example, we note the state-of-the-art ED model (Wadden et al., 2019) can strike 90% in F1 for the type DIVORCE, yet only 50% for the type START-POSITION, and it is more surprising that the training set of DIVORCE is eight times smaller than that of START-POSITION. Finding the causes underlying

[Divorce]
S1: The couple **divorced** four years later.

[Start-Position]
S2: He **became** the first US minister to England.

Figure 1: Two typical event instances of DIVORCE and START-POSITION (taken from the ACE 2005 benchmark), where the trigger words are colored.

the skewed performance is crucial to the robustness of an ED model; however, this problem is still understudied in current research.

In this study we take a fresh look at above problem and for the first time attribute the skewed performance to the *contextual patterns of events*. Let consider the two typical instances of DIVORCE and START-POSITION shown in Figure 1. Intuitively, they demonstrate distinct patterns: the DIVORCE event is more *trigger-dependent*, and the trigger word (i.e., "divorced") is very indicative of the event's occurrence; by contrast, the START-POSITION event is more *context-dependent* — the event semantic is primarily expressed by contexts rather than the trigger "become", which is a merely light verb. We hypothesize an ED model performs poorly on context-dependent types because capturing context semantics is challenging (Lu et al., 2019; Liu et al., 2020b). With the above intuitions, two questions rise: (i) Can we estimate an event's pattern quantitatively? (ii)) How to robustify an ED model by characterizing such patterns?

To address the first question, we introduce a brandy new concept called *trigger saliency attribution*, which can explicitly quantify an event's contextual pattern. Figure 2 illustrates the key idea: to determine how much an event is trigger-dependent or context-dependent, we measure the trigger's contribution to expressing overall the event semantic. Specifically, we first assign each sentence a global event label that represents the overall event semantic. Then, inspired by the feature attribution method

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4573 - 4585

(Simonyan et al., 2014; Sundararajan et al., 2017), we regard each word as a feature and compute its contribution (i.e., saliency value) for predicting the global event label. Finally, by examining the ground-truth trigger's saliency value, we can tell how much an event depends on triggers or contexts: a higher value, for example, indicates that the trigger contributes more to the event, implying the event is more trigger-dependent.

To answer the second question, we develop a new training mechanism based on trigger saliency attribution, which uses saliency as evidence to enhance learning. Our method is simple and straightforward — instead of using a single model to detect all event types, we group event types with similar patterns together (assessed by trigger saliency attribution) and develop separate models for each group. This strategy enables different models to capture distinct patterns — for example, the model for context-dependent type can focus on mining contextual information for learning. To further boost learning, we also propose two saliency-exploration strategy to augment the above framework, which can explicitly integrate saliency information into learning and produce improved performance particularly for context-dependent types (§ 6.2).

To verify the effectiveness of our approach, we have conducted extensive experiments on two ED benchmarks (i.e., ACE 2005 (LDC, 2005) and MAVEN (Wang et al., 2020)). According to the results: (i) Our trigger saliency attribution method can capture the underlying pattern and well explain the skewed performance, obtaining Spearman's correlation coefficients of 0.72 and 0.61 with per-type F1 on ACE 2005 and MAVEN respectively; (ii) Our new training regime based on saliency demonstrates improved results on the two benchmarks. On ACE 2005, for example, it produces a 2% absolute gain in F1 over methods training different event types jointly. Finally, in ablation studies, we compare and highlight many significant characteristics (e.g., linguistic and lexical patterns) of trigger-dependent and context-dependent event types; our work may inspire future research into their patterns.

To summarize, our contributions are three-fold:

- We analyze the origins of an ED model's skewed performance and propose a new notion termed trigger saliency attribution, which can assess the underlying pattern of events. Our findings, as a seminal study, raises the possibility that the traditional "one model fits all types" paradigm may need to be changed.

- We present a new ED training mechanism based on trigger saliency attribution that achieves promising results on two benchmarks, especially when dealing with context-dependent event types.

- We highlight several diverse patterns of trigger-dependent and context-dependent event types, and our findings may stimulate future research into their differences.

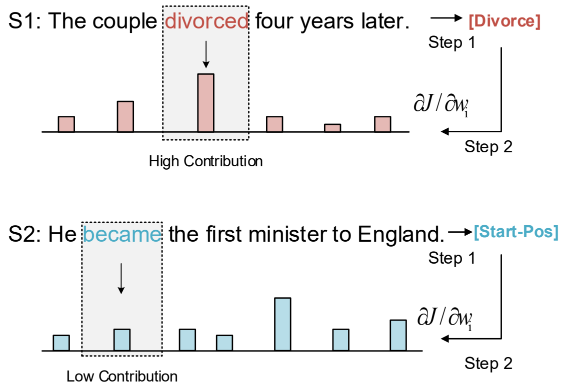

Figure 2: Illustration of trigger saliency attribution, where the saliency value of a trigger can quantify its contribution to the overall event semantic.

2 Background and Related Work

Event Detection. ED is a critical subtask of event extraction that seeks to locate event instances in text, which has received a lot of attention from researchers. Traditional methods for ED typically use fine-grained features (Ahn, 2006; Ji and Grishman, 2008; Liao and Grishman, 2010; Hong et al., 2011; Li et al., 2013), whereas newer methods rely on neural networks (Chen et al., 2015; Nguyen and Grishman, 2015; Feng et al., 2016; Nguyen and Nguyen, 2019; Liu et al., 2018a, 2019a,b), which have investigated the use of syntactic information (Liu et al., 2018b; Lai et al., 2020), document-level cues (Wadden et al., 2019; Lin et al., 2020; Du and Cardie, 2020; Liu et al., 2020b; Lai et al., 2021; Pouran Ben Veyseh et al., 2021; Li et al., 2021; Chen et al., 2021; Liu et al., 2021), and external supervision signals (Tong et al., 2020; Liu et al., 2020a) to boost learning. However, most methods recognize no distinction between event types and train a single model to identify all event types, resulting in rather skewed performance on different event types. Two seminal works (Lu et al., 2019;

Liu et al., 2020b) have observed the comparatively poor performance on context-dependent texts and offered a better context-exploration strategy to improve training. Nonetheless, they are in a position to improve performance rather than investigate the root causes. Our approach, on the other hand, takes a fresh look at the issue and aims to define the underlying patterns of events for learning.

Feature Attribution. The goal of feature attribution (FA) is to assess how important an input feature for model prediction, which has sparked a lot of interest in interpreting model decisions (Simonyan et al., 2014; Sundararajan et al., 2017). Formally, suppose we have an input vector $x = (x_1, x_2, ..., x_n) \in \mathbb{R}^n$ and a function $\mathcal{F}$: $\mathbb{R}^n \rightarrow [0, 1]$ representing a model. The attribution value of x, with respect to the output $\mathcal{F}(x)$, is defined as a vector $A_{\mathcal{F}}(x) = (a_1, a_2, ..., a_n) \in \mathbb{R}^n$, where a_i measures the contribution of x_i to $\mathcal{F}(x)$. The existing FA methods are classified as gradient-based methods, which consider the gradient of the output to the input as the attribution value (Simonyan et al., 2014; Springenberg et al., 2015), and reference-based methods, which consider the difference between the model's output and some "reference" output, in terms of the difference between the input and some "reference" input, as the attribution value (Ribeiro et al., 2016; Sundararajan et al., 2017). FA have been used to interpret model predictions in applications including image classification (Simonyan et al., 2014), machine translation (Ding et al., 2017), text classification (Chen et al., 2018), and others (Bastings and Filippova, 2020). To the best of our knowledge, this is the first work introducing FA to ED for quantifying the underlying event patterns.

Integrated Gradient. Integrated Gradient (Sundararajan et al., 2017) is a specific (reference-based) FA method that views the feature attribution value as the accumulated gradient along the line between the model's input x and a reference input x', which denotes the lack of a feature[1]. Particularly, the attribution value of x_i (i.e., the i^{th} dimension of x) with respect to an output $\mathcal{F}(x)$ is defined as:

$$a_i = (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial \mathcal{F}(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha \quad (1)$$

where $\frac{\partial \mathcal{F}(x)}{\partial x_i}$ indicates the gradient of $\mathcal{F}(x)$ to x_i. In our approach, we prefer Integrated Gradient to other FA methods due to its computing efficiency and effectiveness in addressing a wide range of text based tasks (Sundararajan et al., 2017; Liu and Avci, 2019; Bastings and Filippova, 2020).

[1] In text related tasks, x' is usually set as a sequence of embedding vectors with all zero values (Wallace et al., 2019).

Algorithm 1: Trigger Saliency Attribution

```
Input :Training set 𝒟; a re-defined event type set 𝒯
▷ Train a sentence-level classifier on 𝒟
for each training instance s ∈ 𝒟 do
    ▷ Conduct sentence-level classifcation on s;
    for each word w_i ∈ s and each type T ∈ 𝒯 do
        ▷ Evalaute word-level saliency with Eq. (4);
    end for
end for
for each event type T ∈ 𝒯 do
    ▷ Evaluate type-level saliency with Eq. (5);
end for
```

3 Trigger Saliency Attribution

Algorithm 1 provides an overview of our trigger saliency attribution method, which consists of three major steps: (i) sentence-level event classification, (ii) word-level saliency estimation, and (iii) type-level saliency estimation. Let $s = [w_1, w_2, \cdots, w_N]$ be a sentence of N words, and the ED task corresponds to predicting an event label sequence $Y_s = [y_1, y_2, \cdots, y_N]$, where $y_i \in \mathcal{T} \cup \{\text{O}\}$ indicates the event label of w_i, $\mathcal{T}$ is a set containing all pre-defined event types, and O is a "null type" denoting no-trigger words.

Sentence-Level Event Classification. We start by giving s a sentence-level event label $\mathcal{G}_s$, which represents the overall event semantic. Let the label be $\mathcal{G}_s = [g_1, g_2, ..., g_{|\mathcal{T}|}] \in \mathbb{R}^{|\mathcal{T}|}$, where $g_i \in \{0, 1\}$ indicates whether a trigger of the i^{th} event type is contained by s (g_i=1) or not (g_i=0). Following that, we construct a sentence-level event classifier and aim to learn a mapping from s to $\mathcal{G}_s$. Particularly, we devise a BERT based sentence classifier (Devlin et al., 2019) and adopt a multi-label binary cross-entropy loss for optimization:

$$\mathcal{L}(\mathcal{G}_s; \boldsymbol{X}_s) = -\frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} g_i \cdot \log(\boldsymbol{o}_i^s) + (1 - g_i) \cdot \log(1 - \boldsymbol{o}_i^s) \quad (2)$$

where $\boldsymbol{X}_s$ is the input embedding of s in BERT, $\boldsymbol{o}^s \in \mathbb{R}^{|\mathcal{T}|}$ indicates the logits vector computed by the classier, and $\boldsymbol{o}_i^s$ denotes the i^{th} element of $\boldsymbol{o}^s$.

Word-Level Saliency Estimation. Based on the sentence-level classifier, we next use Integrated Gradient (Sundararajan et al., 2017) to calculate the contribution (i.e., saliency value) of each word

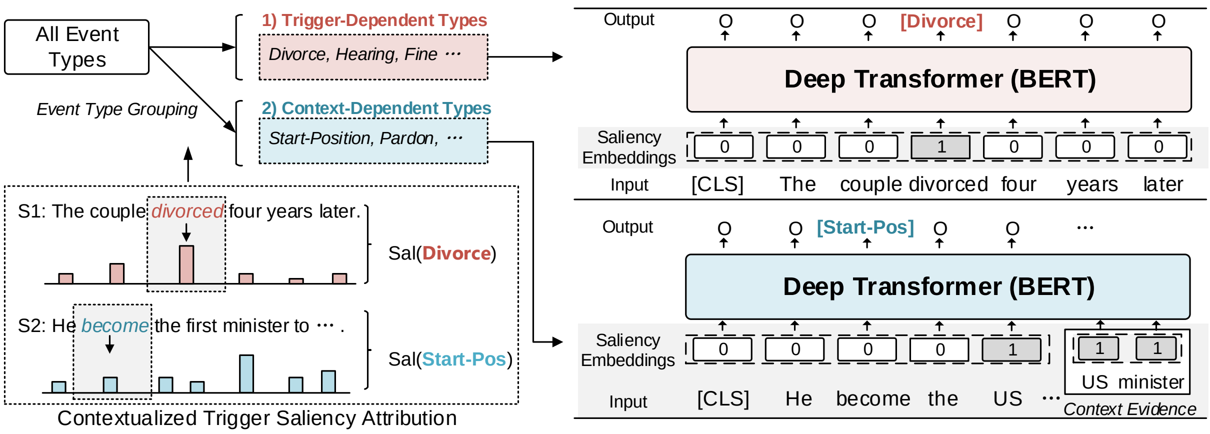

Figure 3: The overview of our saliency enhanced ED model; it first divides event types into different sets based on their patterns and then uses separate models, with different saliency-exploration strategies, to address each set.

to the prediction. We utilize the loss function as the desired model (Wallace et al., 2019), and calculate the saliency of w_i, more accurately, its BERT representation $\boldsymbol{x}_i \in \boldsymbol{X}_s$, regarding the loss by:

$$\boldsymbol{\alpha}_{w_i} = (\boldsymbol{x}_i - \boldsymbol{x}'_i) \times \int_{\alpha=0}^{1} \frac{\partial \mathcal{L}(\mathcal{G}_s; \boldsymbol{X}' + \alpha \times (\boldsymbol{X}_s - \boldsymbol{X}'))}{\partial \boldsymbol{x}_i} d\alpha \quad (3)$$

where $\boldsymbol{X}'$ is a sequence of all-zero vectors (serving as a reference input), and $\boldsymbol{x}'_i$ denotes the i^{th} element in $\boldsymbol{X}'$. We then normalize $\boldsymbol{\alpha}_{w_i}$ as a scalar value α_{w_i} with a sentence-wise normalization:

$$\alpha_{w_i} = e^{\|\boldsymbol{\alpha}_{w_i}\|_2} / \sum_{n=1}^{N} e^{\|\boldsymbol{\alpha}_{w_n}\|_2} \quad (4)$$

where $\|\|$ denotes the L_2 norm. In actuality, we may not be concerned with a word's saliency to the general event semantic $\mathcal{G}_s$, but rather with a specific event type $T \in \mathcal{T}$. To this end, we replace $\mathcal{G}_s$ with the one-hot representation of T in Equation (3) for evaluation. Finally, we represent the word-level saliency of w_i with respect to the event type T by $\alpha_{w_i}^{(T)}$, and we suppose $\alpha_{w_i}^{(T)} = 0$ if the sentence does not describe any event of type T.

Type-Level Saliency Estimation. Based on the word-level saliency, we measure the type-level trigger saliency value (regarding an event type T) as:

$$\text{SL}(T) = \frac{\sum_{(s,Y_s)} \sum_{w \in \{w_i | y_i = T\}} \alpha_w^{(T)}}{\#\,\text{of training examples of type}\, T} \quad (5)$$

where (s, Y_s) ranges over each training instance; $\{w_i | y_i = T\}$ is a set containing all of the triggers of type T in s. The type-level saliency vale $\text{SL}(T)$ indicates how trigger-dependent or context-dependent an event type T is, and it has been shown to correlate strongly with the per-type model performance (§ 6.1).

4 Saliency Enhanced ED

Based on trigger saliency attribution, we devise a new training paradigm for ED, which can distinguish event types with similar patterns for learning and achieves promising results. The overview is shown in Figure 3, and the technical details follow.

Event Type Division. Based on type-level saliency estimation, we divide all event types into a trigger-dependent set $\mathcal{T}_{\text{trigger}} = \{T | \text{SL}(T) \geq \lambda\}$ and a context-dependent set $\mathcal{T}_{\text{context}} = \{T | \text{SL}(T) < \lambda\}$. The threshold λ is empirically determined as the median of all per-type trigger saliency values, implying that the event types are evenly divided into two sets[2].

Saliency-Enriched Event Detector. Following that, we create separate ED models for $\mathcal{T}_{\text{trigger}}$ and $\mathcal{T}_{\text{context}}$. Each model is implemented using the BERT architecture (Devlin et al., 2019), and given a sentence s, it performs a word-by-word classification over BERT's output to generate a label sequence: $\tilde{Y}_s = (\tilde{y_1}, \tilde{y_2}, \cdots, \tilde{y_N})$, with $\tilde{y_i}$ being the predicted event label for w_i. Based on the different characteristics of trigger-dependent and context-dependent types, we devise different saliency-exploration methods to boost learning.

(i) Word Saliency Embeddings. Given that trigger-dependent types often have indicative trig-

[2] We have tried using more than two sets for division in our pilot experiments, but the results were negative.

gers, we build a mechanism called word saliency embeddings (WSEs) in the model for $\mathcal{T}_{\text{trigger}}$ to capture such regularities. Specifically, we first quantify each word's saliency value[3] as 0 or 1 based on λ, i.e., the threshold we used previously for distinguishing event types, and then use a separate embedding vector to distinguish 0 and 1, similar to word embeddings. Such embeddings are incorporated into the model[4] to capture a regularity that words with high saliency values are more likely to be triggers. Note WSEs are also incorporated in the model for the $\mathcal{T}_{\text{context}}$, which on the other hand seeks to learn the opposite regularity that words with high saliency values may not be triggers.

(ii) Saliency as Context Evidence. In the event detector for $\mathcal{T}_{\text{context}}$, we also devise a regime for interpreting salient information as context evidence for reasoning. Consider the previous example S2. Our method identifies the context words "US minister" as the most salient words (with saliency values larger than λ) expressing the overall event semantic. Here we regard salient contexts as supplementary evidence and concatenate them with the sentence for learning, as shown in the bottom of Figure 3. Compared with WSEs, this method can additional capture the lexical semantics of the salient words, which has been shown to considerably aid in the recognition of context-dependent event types (§ 7).

Model Ensemble. In the testing stage, we combine the results of two models to make a final prediction. If ambiguous cases occur, i.e., the two ED models predict different event types for the same word, we use the type with a higher probability as the result. We use cross-entropy loss for optimization. For example, the model for $\mathcal{T}_{\text{trigger}}$ is trained by minimizing the following loss:

$$\mathcal{L} = -\sum\nolimits_{(s,Y_s)} \sum\nolimits_{(w_i,y_i)\in(s,Y_s)} \log P(y_i|w_i) \quad (6)$$

where (s, Y_s) refers to each training instance; (w_i, y_i[5]) ranges over each pair of word and its ground-truth event label; $P(y_i|w_i)$ denotes the conditional probability that the model predicts y_i for w_i. We use Adam (Kingma and Ba, 2015) with default hyper-parameters for parameter update.

[3]To prevent label leaking, at the testing stage we use predicted labels rather than ground-truth labels for attribution.

[4]Because combining external embeddings with BERT remains difficult, we alter the segmentation embeddings in BERT to WSEs, motivated by (Wu et al., 2019).

[5]Note in the event detector for $\mathcal{T}_{\text{trigger}}$, we should consider y_i as O for $y_i \in \mathcal{T}_{\text{context}}$.

Dataset	# Type	Split	# Sen.	# Tok.	# Trig.
ACE	33	Training	17,172	267,959	4,420
		Dev.	923	18,246	505
		Test	832	19,061	424
MAVEN	168	Training	32,431	832,186	77,993
		Dev.	8,042	204,556	18,904

Table 1: Statistics of ACE 2005 and MAVEN, where # Sen., # Tok., and # Trig. indicate the number of event types, sentences, tokens, and triggers respectively.

5 Experimental Setups

Datasets. We conduct experiments on ACE 2005 (LDC, 2005) and MAVEN (Wang et al., 2020). ACE 2005 defines 33 event types and contains 599 documents. We adopt a common split for evaluation following previous works (Li et al., 2013; Wadden et al., 2019). MAVEN is a newly released corpus defining 168 more fine-grained event types (Wang et al., 2020). Because the MAVEN test set is not publicly available and our study is concerned with per-type performance, we instead use the MAVEN development set for assessment and divide the original MAVEN training set as 9:1 for training and validating. Table 1 displays the comprehensive data statistics for the two datasets.

Evaluation Metrics. We adopt the following metrics to evaluate our model: (i) Spearman's rank correlation coefficient, which can determine the statistical dependency between two ranked variable sequences. The metric is defined as $\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$, where d_i is the difference between the i^{th} pair of ranked variables, and n is the sequence length. We use it to measure how well our trigger saliency attribution results correlate with per-type model performance. (ii) Precision (P), Recall (R) and (Micro) F1, which are widely used to assess the *overall performance* of an ED model. (iii) Macro F1, the arithmetic mean of class-wise F1-scores, which will be low for models that only perform well on common types but badly on rare types.

Implementations. In our trigger saliency attribution method, the sentence-level classifier is built on the `BERT-base`. The batch size is set to 20, and the learning rate is set to 1e-5. After 5 epochs, it achieves 74.8% in F1 on the ACE 2005 development set, matching the state-of-the-art performance (Liu et al., 2019c). As for the two ED models, we consider `BERT-base` architectures. The batch size is set to 20, chosen from [1, 5, 10, 20, 30]. The

Setting	Method	Dataset	
		ACE 05	MAVEN
Static	# of Training Instances	0.06	0.09
	Trigger Variance	0.26	0.25
Dynamic	Trigger Attention	0.12	0.14
	Trigger Saliency (Ours)	**0.72**	**0.61**

Table 2: The Spearman's ρ correlation ($\rho \in$ [-1, 1]) between per-type F1 and different criteria (high correlation is considered when $\rho > 0.6$).

learning rate is set to 1e-5, chosen from a range from 1e-3 to 1e-6. The dimension of word saliency embeddings is empirically set to 100. To allow for further investigation, we have made our code publicly available at `https://github.com/jianliu-ml/SaliencyED`.

6 Experimental Results

6.1 Results of Correlation Measurement

Table 2 shows the Spearman's rank correlation between per-type F1 and four criteria: 1) the number of training instances (regarding an event type); 2) trigger variance, defined as the ratio of the number of unique event triggers to the total number of event triggers (regarding an event type); 3) trigger attention value, which corresponds to the ground-truth trigger's attention value in the BERT model; 4) trigger saliency attribution (our method). We use a state-of-the-art ED model (Wadden et al., 2019) and perform a 5-run average on the development set to obtain the per-type F1 score.

According to the results, our trigger saliency attribution approach correlates the best with model performance, yielding a score as high as 0.72 and 0.61 in Spearman's ρ correlation. This suggests that our method can well explain the skewed performance. Our other findings are interesting: (i) Surprisingly, the number of training examples shows a negligible correlation (ρ = 0.06 and 0.09) with per-type F1. This implies that simply collecting more training data may not be an effective way to improve an ED model. (ii) The trigger variance metric demonstrates a moderate association (ρ = 0.25 and 0,26), indicating that the diversity of event triggers is a factor influencing model performance. (iii) The trigger attention value also shows a poor association, which may be another proof that attention is not explainable (Jain and Wallace, 2019).

Lastly, Figure 4 visualizes correlations between per-type F1 and the number of training instances

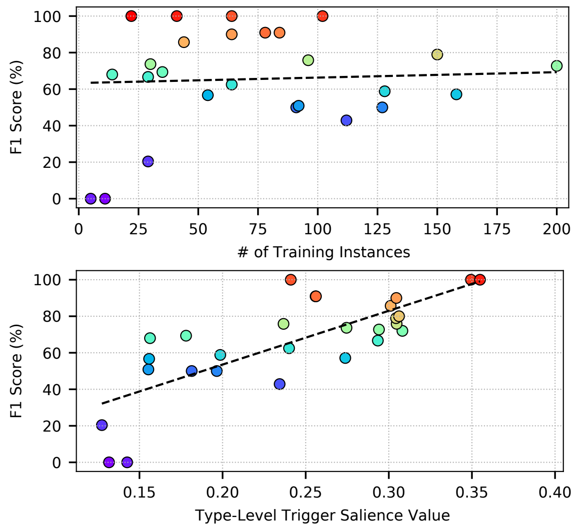

Figure 4: Correlation between per-type F1 and (i) the number of training instances (top), and (ii) type-level trigger salience value (bottom), based on ACE 2005. Each point indicates a specific event type.

and our trigger saliency attribution method. In addition to noting that our method adequately explains the per-type F1-score, we find that $\lambda = 0.25$ may be a good threshold for distinguishing between trigger-dependent and context-dependent event types.

6.2 Results of Saliency Enhanced ED

To test the efficacy of our saliency enhanced ED model: 1) For ACE 2005, we compare our model with (i) DYGIE++ (Wadden et al., 2019), which uses a graph view to learn context features; (ii) TriggerQA (Du and Cardie, 2020), which uses a question answering formulation for the task; (iii) OneIE (Lin et al., 2020), which adopts cross-sentence features for the task. Because pre-processing has a significant impact on the results (Orr et al., 2018), to ensure a fair comparison, we only consider models using the same pre-processing steps as in (Wadden et al., 2019). 2) For MAVEN, we use the BERT+CRF proposed in the original work (Wang et al., 2020) for comparison. As a baseline, we also construct a model called BERTEns, which ensembles two BERT models similar to ours but does not differentiate event types. We refer to our approach that merely separates event types for learning (without saliency-exploration strategies) as SaliencyED (SL), and our full approach as SaliencyED (Full). Table 3 displays performances of different models.

The results have confirmed our approach's effectiveness. Particularly: (i) our full model achieves the best Micro F1 score (75.8% and 67.1%) on

	Method	P ▲	R ▲	F1 ▲	F1 ▽
ACE	DYGIE++ (2019)	-	-	73.6	65.7
	TriggerQA (2020)	71.2	73.7	72.4	64.5
	OneIE (2020)	-	-	75.2	66.6
	BERTEns	71.5	73.1	72.3	65.4
	SaliencyED (SL)	74.7	75.5	75.1	68.1
	SaliencyED (Full)	**75.4**	**76.2**	**75.8**	**68.8**
MAV	BERT+CRF (2020)	62.3	64.1	63.2	55.2
	BERTEns	64.7	66.9	65.8	58.0
	SaliencyED (SL)	**64.9**	68.2	66.5	59.2
	SaliencyED (Full)	**64.9**	**69.4**	**67.1**	**60.3**

Table 3: Results on ACE 2005 and MAVEN (MVN). P ▲, R ▲, and F1 ▲ indicate Precision, Recall, and Micro F1 respectively; F1 ▽ denotes Macro F1.

ACE 2005 and MAVEN without the use of sophisticated architectures or external resources, as DYGIE++ and OneIE do. (ii) Impressively, with the identical architectures, our full model SaliencyED (Full) outperforms BERTEns by 2.8% and 1.7% in F1 on the two datasets, respectively; SaliencyED (SL), which only differentiates event types for training, outperforms BERTEns by 1.6% in F1. This emphasizes the significance of identifying event patterns for ED. (iii) Our method gives the best Macro F1 on two datasets, indicating that it performs well on both common and rare event types.

Table 4 shows the performance breakdown for trigger-dependent (TD) and context-dependent (CD) types. According to the results, different models consistently produce good performance on TD types but low performance on CD types, implying that the patterns found by our trigger saliency attribution method are reasonable. When comparing SaliencyED (SL) and SaliencyED (Full), we see that the saliency-exploring method is more effective on CD types (+2.3% in F1) than on TD types (+0.3% in F1). This makes sense because detecting context-dependent events relies significantly on context reasoning, and our method can just use important contexts as evidence to improve learning.

7 Discussion

Ablation Study. We undertake an ablation study in Table 5 to investigate different model components, using the more challenging context-dependent (CD) types as an example. In the variant models, +WSE and +Evidence denote supplementing SaliencyED (SL) with word saliency embeddings and context evidence, respectively. +MaskAtt is an approach for calculating attention that masks the word itself, which can drive the model to focus more on contexts for learning; +Gold Argument is an oracle method that uses gold event arguments as evidence for learning. Based on the results, +Evidence outperforms +WSE and +MaskAtt, indicating its efficacy. Interestingly, +MaskAtt also boosts performance, implying that the contexts of CD events do carry important information for asserting the event. Finally, the superior performance of +Gold Arguments implies that finding indicative evidence (e.g., event arguments) is the key factor boosting learning on CD types.

		TD Types		CD Types	
	Method	F1 ▲	F1 ▽	F1 ▲	F1 ▽
ACE	DYGIE++ (2019)	78.2	74.4	65.8	52.1
	TriggerQA (2020)	80.1	76.3	65.2	53.2
	OneIE (2020)	83.6	77.9	69.0	54.2
	BERTEns	83.3	77.8	68.3	52.3
	SaliencyED (SL)	86.2	**82.0**	70.0	56.9
	SaliencyED (Full)	**86.4**	81.6	**71.5**	**57.8**
MAV	BERT+CRF (2020)	67.5	67.1	49.2	38.1
	BERTEns	70.3	70.0	51.5	38.1
	SaliencyED (SL)	71.3	70.2	52.6	49.1
	SaliencyED (Full)	**71.6**	**70.8**	**53.5**	**50.4**

Table 4: Results on trigger-dependent (TD) and context-dependent (CD) event types, where F1 ▲ and F1 ▽ indicate Micro and Macro F1 respectively.

Method	F1 ▲	F1 ▽
SaliencyED (SL)	70.0	56.9
SaliencyED (SL) + WSE	70.2	57.3
SaliencyED (SL) + Evidence	**70.6**	**57.5**
SaliencyED (SL) + MaskAtt	70.4	57.1
SaliencyED (Full)	71.5	57.8
SaliencyED (Full) + Gold Arguments	**78.2**	**68.9**

Table 5: Ablations on context-dependent types. F1 ▲ and F1 ▽ indicate Micro and Macro F1 respectively.

Impact of Event Type Division. We use our event type division method as a baseline and compare it to three other event type division strategies: 1) at random; 2) based on the amount of training instances; 3) based on development set performance. According to the results, the first two strategies decrease performance by 1.27% and 1.41% in Micro F1 on ACE, and 1.53% and 1.40% on MAVEN, which suggests that an inappropriate separation of event types impairs learning. The third strategy based on development performance improves learning (+0.8%/+1.1% on ACE/MAVEN), but it

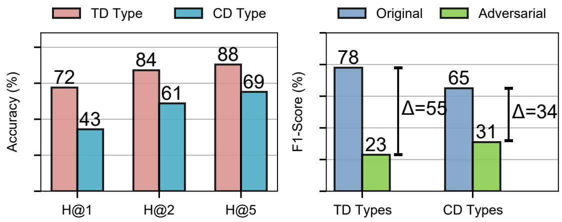

Figure 5: Left: Top k accuracy (hit@k) when the most salient word appears to be an event trigger. Right: Performance drop in an adversarial attack, where the event trigger is masked for sentence-level classification.

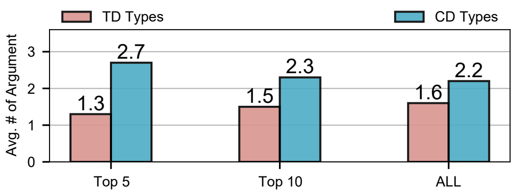

Figure 6: A comparison of the average amount of event arguments in TD and CD types.

8 Most Trigger-Dependent (TD) Types:
Divorce$_{(0.434)}$, Hearing$_{(0.355)}$, Fine$_{(0.349)}$, Injure$_{(0.308)}$, Be_Born$_{(0.306)}$, Elect$_{(0.305)}$, Sentence$_{(0.304)}$, Die$_{(0.304)}$
8 Most Context-Dependent (CD) Types:
Start_Org$_{(0.127)}$, Pardon$_{(0.129)}$, Nominate$_{(0.132)}$, Extradite$_{(0.134)}$, Acquit$_{(0.142)}$, Merge_Org$_{(0.151)}$, Transfer_Money$_{(0.155)}$, End_Org$_{(0.156)}$

Table 6: Typical TD and CD types on ACE 2005.

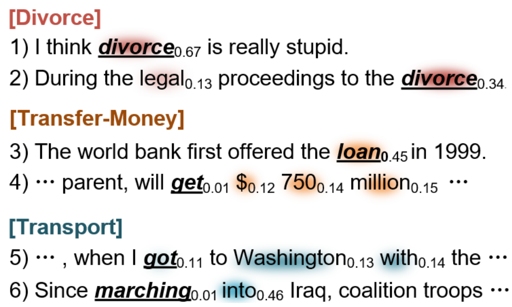

Figure 7: Case visualization, where the ground-truth event triggers are underlined. Color is used to represent words with large saliency values (≥ 0.1).

is still inferior to our approach. An explanation is that the final model performance is the product of a combination of factors, and thus categorizing event types based on development set performance may not assure that event types with similar patterns are grouped together, resulting in inferior results.

Distinctions in TD/CD Types. We use ACE 2005 as a case to highlight the distinct characteristics between TD and CD types. Figure 5 (Left) depicts the top k accuracy (hit@k) in the case where the most salient word in a sentence appears to be an event trigger; Figure 5 (Right) depicts the performance drop in an adversarial attack in which the gold event triggers are masked for sentence-level event type classification. The CD and TD types exhibit opposing behaviors: TD types display excellent H@k accuracy but a significant performance loss in adversarial attack, whereas CD types exhibit the opposite tendency. This implies that the CD and TD types respectively rely on triggers and contexts. Figure 6 shows a comparison of the number of event arguments for TD and CD types. Clearly, CD types have a larger number of event arguments than TD types. This is also another indication that CD types rely on contexts — they require more arguments to convey an event.

Linguistic/Lexical Insights. Table 6 give typical TD and CD types on ACE 2005 (Please refer to Appendixes for the full set). Intuitively, the TD types appear to be finer-grained and concrete, whereas the CD types appear to be coarser-grained and abstract. For example, we may further subdivide a CD type TRANSFER_MONEY into finer-grained ones like LOAN and PURCHASE. We provide linguistic/lexical insights by comparing the hierarchy levels of TD/CD types on WordNet (Miller, 1992). Accordingly, triggers of TD types are at the lower level of WordNet, with an average of 5.6 hypernyms; yet CD type triggers are at a higher level of WordNet, with 2.3 hypernyms. This finding supports our intuition that TD types are more concrete whereas CD types are more abstract.

Case Visualization. Figure 7 depicts the saliency map of several cases. Accordingly, event triggers of TD types do usually have large saliency values. For example, case 2) is the instance of DIVORCE with the lowest trigger saliency value, which is still as high as 0.34. In contrast, event triggers of CD types typically have low saliency values. For example, case 4) and 6) show random instances of TRANSFER-MONEY and TRANSPORT, where the trigger saliency values are only 0.01.

8 Conclusion

In this study, we analyze the origins of an ED model's skewed performance and introduce a new notion called trigger saliency attribution to quantify the pattern of events. We devise a new training paradigm for ED that can distinguish between trigger-dependent and context-dependent types for learning, yielding promising results on two benchmarks. We also examine the differences between the two types extensively, and our work may promote future research on this problem. In the future, we would apply our method to other tasks (e.g., relation extraction) where contextual patterns matter.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.62106016). This work is also supported by Fundamental Research Funds for the Central Universities (No. 2021RC234), the National Key R&D Program of China (2019YFB1405200), and the Open Projects Program of National Laboratory of Pattern Recognition.

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, Sydney, Australia. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. 2018. Learning to explain: An information-theoretic perspective on model interpretation. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 882–891. PMLR.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2021. Honey or poison? solving the trigger curse in few-shot event detection via causal intervention. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8078–8088, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1150–1159, Vancouver, Canada. 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.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. 2016. A language-independent neural network for event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 66–71, Berlin, Germany. Association for Computational Linguistics.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1127–1136, Portland, Oregon, USA. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Viet Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2021. Learning prototype representations across few-shot tasks for event detection. In *Proceedings of* *the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5270–5277, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5405–5411, Online. Association for Computational Linguistics.

LDC. 2005. *ACE (Automatic Content Extraction) English Annotation Guidelines for Events*, 5.4.3 2005.07.01 edition.

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.

Rui Li, Wenlin Zhao, Cheng Yang, and Sen Su. 2021. Treasures outside contexts: Improving event detection via global statistics. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2625–2635, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797, Uppsala, Sweden. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Frederick Liu and Besim Avci. 2019. Incorporating priors with feature attribution on text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6274–6283, Florence, Italy. Association for Computational Linguistics.

Jian Liu, Yubo Chen, and Kang Liu. 2019a. Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6754–6761.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020a. 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.

Jian Liu, Yubo Chen, Kang Liu, Yantao Jia, and Zhicheng Sheng. 2020b. How does context matter? on the robustness of event detection with context-selective mask generalization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2523–2532, Online. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2018a. Event detection via gated multilingual attention mechanism. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4865–4872. AAAI Press.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2019b. Neural cross-lingual event detection with minimal parallel resources. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 738–748, Hong Kong, China. Association for Computational Linguistics.

Jian Liu, Yufeng Chen, and Jinan Xu. 2021. Machine reading comprehension as data augmentation: A case study on implicit event argument extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2716–2725, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shulin Liu, Yang Li, Feng Zhang, Tao Yang, and Xinpeng Zhou. 2019c. Event detection without triggers. 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 735–744, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018b. 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.

Yaojie Lu, Hongyu Lin, Xianpei Han, and Le Sun. 2019. Distilling discrimination and generalization knowledge for event detection via delta-representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4366–4376, Florence, Italy. Association for Computational Linguistics.

George A. Miller. 1992. WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

T. Mitamura, Zhengzhong Liu, and E. Hovy. 2017. Events detection, coreference and sequencing: What's next? overview of the tac kbp 2017 event track. *Theory and Applications of Categories*.

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 (Volume 2: Short Papers)*, pages 365–371, Beijing, China. Association for Computational Linguistics.

Trung Minh Nguyen and Thien Huu Nguyen. 2019. One for all: Neural joint modeling of entities and events. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6851–6858. AAAI Press.

Walker Orr, Prasad Tadepalli, and Xiaoli Fern. 2018. Event detection with neural networks: A rigorous empirical evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 999–1004, Brussels, Belgium. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Minh Van Nguyen, Nghia Ngo Trung, Bonan Min, and Thien Huu Nguyen. 2021. Modeling document-level context for event detection via important context selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5403–5413, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marco Túlio 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, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144. ACM.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*.

J.T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. 2015. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328, International Convention Centre, Sydney, Australia. PMLR.

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, Online. Association for Computational Linguistics.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. AllenNLP interpret: A framework for explaining predictions of NLP models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 7–12, Hong Kong, China. Association for Computational Linguistics.

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 (EMNLP)*, pages 1652–1671, Online. Association for Computational Linguistics.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *International Conference on Computational Science*, pages 84–95. Springer.

A The Full Event Types and Their Saliency Values

We provide the full set of event types in ACE (LDC, 2005) and MAVEN (Wang et al., 2020) and their saliency values evaluated by our method.

Trigger-Dependent Types		Context-Dependent Types	
Divorce	0.434	Demonstrate	0.239
Trial_Hearing	0.354	Attack	0.236
Fine	0.349	Phone_Write	0.234
Injure	0.308	End_Position	0.198
Be_Born	0.306	Start_Position	0.196
Elect	0.304	Transfer_Ownership	0.181
Sentence	0.304	Execute	0.178
Die	0.304	Meet	0.178
Marry	0.301	Transport	0.156
Appeal	0.294	End_Org	0.155
Declare_Bankruptcy	0.293	Transfer_Money	0.155
Charge_Indict	0.274	Merge_Org	0.150
Sue	0.273	Acquit	0.142
Arrest_Jail	0.256	Extradite	0.134
Convict	0.255	Nominate	0.131
Release_Parole	0.241	Pardon	0.128
		Start_Org	0.127

Table 7: Event types and their trigger saliency values in the ACE ontology.

Trigger-Dependent Types		Context-Dependent Types	
Commerce_sell	0.221	Cause_to_make_progress	0.104
Rescuing	0.195	Cost	0.104
Use_firearm	0.168	Hold	0.103
Receiving	0.165	Award	0.102
Becoming	0.160	Check	0.102
Bodily_harm	0.160	Being_in_operation	0.101
Choosing	0.159	Manufacturing	0.101
Destroying	0.157	Bringing	0.100
Escaping	0.156	Response	0.099
Death	0.152	Know	0.099
Arranging	0.150	Perception_active	0.098
Cause_change_of_strength	0.150	Ratification	0.097
Competition	0.150	Creating	0.096
Defending	0.146	Prison	0.096
Besieging	0.146	Testing	0.096
Expressing_publicly	0.146	Incident	0.092
Conquering	0.145	Kidnapping	0.092
Surrendering	0.144	Legal_rulings	0.089
Arrest	0.144	Temporary_stay	0.088
Dispersal	0.143	Imposing_obligation	0.087
Sending	0.143	Scouring	0.086
Control	0.143	Social_event	0.086
Preserving	0.142	Motion	0.085
Influence	0.140	Create_artwork	0.084
Commerce_buy	0.138	Action	0.082
Coming_to_be	0.137	Collaboration	0.078
Damaging	0.136	Come_together	0.078
Earnings_and_losses	0.135	Robbery	0.077
Motion_directional	0.135	Scrutiny	0.076
Assistance	0.135	GetReady	0.076
Killing	0.134	Legality	0.076
Commerce_pay	0.131	Emptying	0.075
Arriving	0.131	Communication	0.075
Deciding	0.131	Coming_to_believe	0.075
Request	0.130	Connect	0.072
Recording	0.129	Forming_relationships	0.071
Supporting	0.128	Institutionalization	0.071
Becoming_a_member	0.128	Reveal_secret	0.067
Aiming	0.127	Patrolling	0.067
Containing	0.125	Rewards_and_punishments	0.065
Name_conferral	0.124	Filling	0.065

Change_event_time	0.124	Self_motion	0.064
Using	0.124	Adducing	0.063
Building	0.124	Cure	0.063
Sign_agreement	0.124	Submitting_documents	0.063
Reporting	0.124	Criminal_investigation	0.063
GiveUp	0.123	Reforming_a_system	0.062
Getting	0.121	Expend_resource	0.062
Recovering	0.120	Rite	0.062
Cause_to_amalgamate	0.118	Commitment	0.061
Cause_to_be_included	0.117	Protest	0.059
Departing	0.117	Statement	0.059
Publishing	0.117	Hiding_objects	0.059
Change	0.117	Limiting	0.058
Agree_or_refuse_to_act	0.117	Committing_crime	0.058
Cause_change_of_position_on_a_scale	0.116	Education_teaching	0.056
Judgment_communication	0.116	Terrorism	0.055
Process_end	0.116	Employment	0.053
Wearing	0.116	Military_operation	0.052
Traveling	0.115	Telling	0.052
Releasing	0.115	Theft	0.050
Giving	0.115	Confronting_problem	0.046
Process_start	0.115	Practice	0.046
Quarreling	0.115	Revenge	0.045
Exchange	0.115	Convincing	0.044
Presence	0.114	Renting	0.043
Preventing_or_letting	0.113	Having_or_lacking_access	0.041
Attack	0.113	Resolve_problem	0.040
Catastrophe	0.112	Labeling	0.038
Hindering	0.111	Vocalizations	0.036
Warning	0.111	Body_movement	0.036
Participation	0.111	Breathing	0.035
Achieve	0.110	Ingestion	0.035
Violence	0.109	Research	0.033
Placing	0.109	Lighting	0.033
Causation	0.108	Justifying	0.032
Hostile_encounter	0.108	Writing	0.032
Surrounding	0.108	Extradition	0.031
Carry_goods	0.107	Suspicion	0.031
Change_of_leadership	0.107	Change_sentiment	0.030
Removing	0.106	Bearing_arms	0.019
Supply	0.105	Change_tool	0.012
Expansion	0.105	Emergency	0.010
Openness	0.105	Risk	0.010

Table 8: Event types and their trigger saliency values in the MAVEN ontology.