How Does Context Matter? On the Robustness of Event Detection with Context-Selective Mask Generalization

Jian Liu[1,2,3], Yubo Chen[1,2], Kang Liu[1,2], Yantao Jia[4], Zhicheng Sheng[4]
[1] National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing, 100190, China
[2] University of Chinese Academy of Sciences
[3] Beijing Jiaotong University, 100044, China
[4] Huawei Technologies Co., Ltd, Beijing, 100085, China
{jian.liu, yubo.chen, kliu}@nlpr.ia.ac.cn; jamaths.h@163.com; shengzhicheng@huawei.com

Abstract

Event detection (ED) aims to identify and classify event triggers in texts, which is a crucial subtask of event extraction (EE). Despite many advances in ED, the existing studies are typically centered on improving the overall performance of an ED model, which rarely consider the robustness of an ED model. This paper aims to fill this research gap by stressing the importance of robustness modeling in ED models. We first pinpoint three stark cases demonstrating the brittleness of the existing ED models. After analyzing the underlying reason, we propose a new training mechanism, called context-selective mask generalization for ED, which can effectively mine context-specific patterns for learning and robustify an ED model. The experimental results have confirmed the effectiveness of our model regarding defending against adversarial attacks, exploring unseen predicates, and tackling ambiguity cases. Moreover, a deeper analysis suggests that our approach can learn a complementary predictive bias with most ED models that use full context for feature learning.

1 Introduction

Event detection (ED), a crucial subtask of event extraction (EE), aims to identify and categorize event triggers in texts. For example, in a sentence S1: "During a war, invaders *destroyed* the whole town", ED requires a system to detect an event trigger *destroyed*, along with its event type ATTACK[1]. Building a robust ED system is shown to benefit a wide range of applications including document summarization (Filatova and Hatzivassiloglou, 2004), knowledge base population (Ji and Grishman, 2011; Mitamura et al., 2017), question answering (Berant et al., 2014), and others.

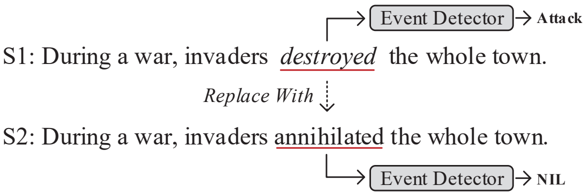
Event Detector
Attack
S1: During a war, invaders destroyed the whole town.
Replace With
S2: During a war, invaders annihilated the whole town.
Event Detector
NIL

Figure 1: Example of adversarial attack in ED.

In recent years, great advances have been made in ED (Ji and Grishman, 2008; Li et al., 2013; Chen et al., 2015; Nguyen et al., 2016; Feng et al., 2016; Liu et al., 2018b,a, 2019b). However, the vast majority of existing studies focus on improving the overall performance of an ED model (usually on a fixed test set), which rarely consider the robustness (and generalization capability) of an ED model. For example, most of existing methods do not answer questions such as when/why an ED system would fail, how to handle new, previously unseen data, despite these considerations are especially crucial for designing real-world ED systems.

This paper focuses on the robustness aspect of ED models. We first emphasize the necessity of this research by pinpointing three stark cases demonstrating the vulnerability of existing ED models. These cases are: 1) adversarial attack, which refers to adding small perturbations in the original sentences (Papernot et al., 2016; Alzantot et al., 2018). As shown in Figure 1, a well-trained event detector can correctly recognize the event trigger *destroyed* at first. But when we replace *destroyed* with a rare trigger *annihilated*, despite the meaning of the sentence does not change, we note the same event detector fails to identify the trigger. A quantitative evaluation suggests that the performance of a state-of-the-art (SoTA) ED model (Chen et al., 2015) drops significantly from 69.1% to 19.2% facing adversarial attack. 2) Unseen predicates, which measures whether an ED model can tackle new, previously unseen data. We note the existing ED models demonstrate a rather poor generalization

[1] According to ACE event ontology.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2523–2532
November 16 - 20, 2020. ©2020 Association for Computational Linguistics

capability: they achieve only 14.2% in F1 for the previously unseen triggers, despite 74.9% in F1 for the already seen triggers. 3) Event type disambiguation, which refers to assign a correct event type to ambiguous triggers, considering that over 70% of triggers can express different types of events (Liu et al., 2018b). While, our pilot experiments suggest that a SoTA ED model obtains only 50.4% in F1 in tackling the high-ambiguity cases, comparing to 70.6% in F1 in tackling the low-ambiguity cases.

The above phenomena reflect the fact that current ED models have a poor ability in modeling *contexts*, where the underlying reason may highly relate to *reasoning shortcuts* (Jiang and Bansal, 2019) — owing to the limited (and biased) training data, an ED model may have only learned lexical pattern, i.e., word-to-trigger mapping (such as *destroyed* → `Attack`), owing to its prevalence in data. By adopting such reasoning shortcuts, an ED model may explain the training data well, but fail in the more context-dependent scenarios noted above, as they never capture the underlying regularities about how event triggers appear in texts.

In light of the above analysis, we propose a new training paradigm, termed as context-selective mask generalization, aiming to prevent reasoning shortcuts and robustify an ED model. Our method is intuitive and straightforward: To prevent lexical bias, we explicitly *delexicalize* triggers for training/testing, by replacing them with placeholders. This forces our model to make predictions using contexts *solely*. For instance, a training example S1 is transferred as: "During a war, invaders [MASK] the whole town", and our model is forced to predict the event label of the masked word. As the lexical information of the trigger is completely masked, our model has to mine the more essential contextual clues for reasoning. This prevents our model simply remembering word-to-trigger shortcuts, but to learning the underlying regularities regarding how events are described in texts.

The proposed learning paradigm consists of two complementary training objectives: **context-selective discriminative learning** and **contextualized similarity learning**. The former is an intra-sentence objective, considering that contextual words are usually of different importance, for example, in S1, "wars" and "invaders" may be more important than "town" for predicting the `Attack` event. We devise a method combing selective attention (Lin et al., 2016) with model uncertainty (Gal and Ghahramani, 2016) to weigh contexts and select the salient parts for learning. The latter is an inter-sentence objective, with an assumption that: *event triggers have same types may occur in similar contexts*, derived from the well-known distributional hypothesis of words (Harris, 1954). We take in pairs of mask-containing sentences as input, and encourage their contextual representations to be similar if the masked triggers express the same type of events.

To verify the effectiveness of our approach, we have conducted extensive experiments on the benchmark event dataset, and we show the definite advantages of our approach over previous methods with respect to: 1) defending against adversarial attack, 2) tackling unseen predicates, and 3) handling ambiguity cases. Moreover, a deeper analysis suggests that our approach can learn a complementary predictive bias with the existing ED models using full context for reasoning.

Contributions. 1) In this work, we stress the importance of robustness modeling in ED, a problem less studied in the existing literature. We pinpoint three stark cases demonstrating the brittleness of existing ED methods, with qualitative evaluation, and analyze the underlying reason. 2) We propose a new training paradigm, called context-selective mask generalization, which can effective mine context-specific patterns for ED, shedding lights on building ED systems of decent robustness. 3) We report on extensive experiments demonstrating the advantages of our model in defending against adversarial attack, handling unseen predicates, and tackling ambiguous cases. We also give a deeper analysis exploring the predictive bias of our method.

2 Related Work

2.1 Event Detection

ED is a crucial subtask of EE that aims to find event triggers in texts. Earlier approaches for ED are feature based. To name a few, Ahn (2006) exploited lexical, syntactic, and external knowledge based features for the task; Ji and Grishman (2008) combined global and local decision features for the task. Liao and Grishman (2010) and Hong et al. (2011) investigated cross-event/cross-entity inference for the task; Li et al. (2013) proposed a joint framework for the task. Modern approaches for ED are neural network based. For example, Chen et al.

(2015) leveraged Convolutional Neural Networks (CNNs) for the task; Nguyen et al. (2016) used Recurrent Neural Networks (RNNs) for the task; Feng et al. (2016) combined CNNs with RNNs and Liu et al. (2018b) explored Graph Convolutional Networks (GCNs) for the task. More recent works have designed advanced architectures for the task (Liu et al., 2017, 2018a; Lu et al., 2019; Liu et al., 2019a).

Despite many advances in ED, to date rare work has studied the robustness (and generalization capability) of an ED model. The work of Lu et al. (2019) is related to ours, which improved the generalization of an ED model by decoupling lexical-specific and lexical-free representations via adversarial training. Compared to their work, the introduction of placeholders in our work can naturally decouple lexical-specific and lexical-free representations, which avoids the unstable adversarial learning process. Moreover, our work evaluates three aspects of robustness, rather than only unseen predicates. Our work also relates to the study of Huang et al. (2018), which aims to recognize events of never-seen event types, i.e. zero-shot EE. Their work lies in an orthogonal dimension of our work regarding the generalization of ED models.

2.2 Robustness Probing in Natural Language Processing Applications

Enhancing the robustness of a model is a challenging and long-standing goal of AI research community. In computer vision, Szegedy et al. (2014) first pointed out that a crafted input with small perturbations could easily fool a neural model, referring to it as *adversarial example*. Papernot et al. (2016) first studied adversarial example in texts, and they proposed to producing adversarial input sequences on Recurrent Neural Network (RNN). Following the work, Alzantot et al. (2018) proposed a population-based optimization method to generate more semantically similar adversarial examples. Many researchers have investigated robustness modeling in specific NLP problems. To name a few, Jia and Liang (2017) inserted adversarial perturbations into paragraphs for machine reading comprehension (MRC). The work was further extended by Mudrakarta et al. (2018), which cast the generation of adversarial examples as an optimization problem for the task of natural language inference (NLI); Belinkov and Bisk (2017); Ebrahimi et al. (2018) investigated how to tackle adversarial examples in neural machine translation (NMT). A very recent work of Hsieh et al. (2019) investigated the robustness of self-attentive architectures (Vaswani et al., 2017) in sentiment analysis, entailment and machine translation under adversarial attacks. But to our best knowledge, there is no work systematically studying the robustness of ED.

3 Approach

Figure 2 visualizes the overview of our approach, by taking S1 as an example. Let a sentence of N words be $S = [w_1, w_2, ..., w_N]$. Following previous works (Li et al., 2013; Chen et al., 2015; Nguyen et al., 2016; Lu et al., 2019), we formulate the ED task as a token-level classification problem. That is, for each word in S, we consider it as a **candidate trigger**, and our goal is to assign a correct event label to it (A type of NIL is used to indicate a non-trigger word).

The technical details of our approach are presented in the following, including: trigger delexicalization (§ 3.1), context-selective discriminative learning (§ 3.2), contextualized similarity learning (§ 3.3), attentive representation fusion (§ 3.4), and the training strategy (§ 3.5).

3.1 Trigger Delexicalization

Following recent advances in ED (Yang et al., 2019), we adopt BERT architecture (Devlin et al., 2019) to learn the input representations, by first adding special tokens at the both ends of S to construct an extended sequence "[CLS] S [SEP]". Note we do not allow our model to leverage lexical clues, we explicitly delexicalize the candidate trigger, by replacing it with a placeholder [MASK]. Consider S1 and S2 in Figure 1. If we take *destroyed* or *annihilated* as the candidate trigger, the mask-containing sequence is "[CLS] During a war, invaders [MASK] the whole town [SEP]". Next. we use BERT for sequence encoding and take the final hidden layer[2] of BERT as the input representations, denoted as $\boldsymbol{H}_S \in \mathcal{R}^{(N+2)\times d}$. We use $\boldsymbol{h}_{w_i} \in \mathcal{R}^d$ to denote the representation of a specific token w_i.

[2] In case a word may be split into many sub-word pieces, we conduct a self-attentive computation over sub-word pieces to compute the representation of original word, as suggested by Lee et al. (2017).

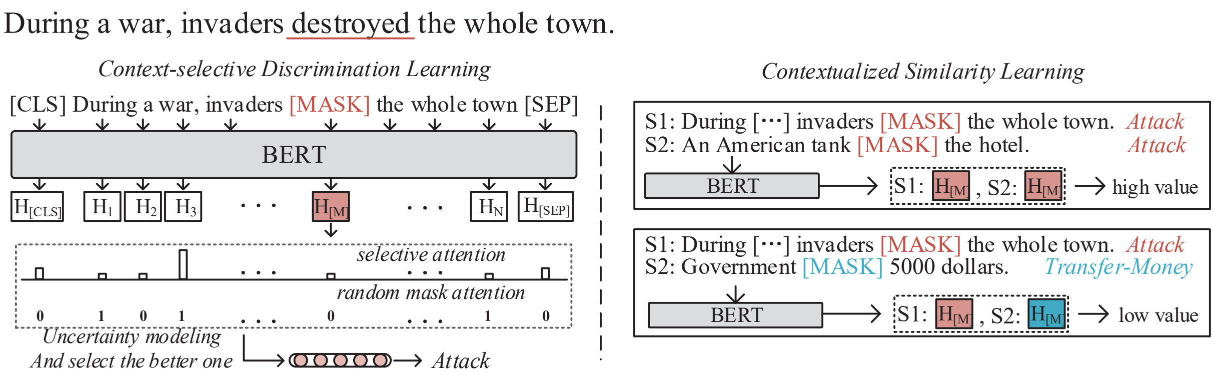
During a war, invaders destroyed the whole town.
Context-selective Discrimination Learning
[CLS] During a war, invaders [MASK] the whole town [SEP]
BERT
H[CLS] H1 H2 H3 · · · H[M] · · · HN H[SEP]
selective attention
random mask attention
0 1 0 1 · · · 0 · · · 1 0
Uncertainty modeling
And select the better one
Attack
Contextualized Similarity Learning
S1: During [···] invaders [MASK] the whole town. Attack
S2: An American tank [MASK] the hotel. Attack
BERT
S1: H[M], S2: H[M]
high value
S1: During [···] invaders [MASK] the whole town. Attack
S2: Government [MASK] 5000 dollars. Transfer-Money
BERT
S1: H[M], S2: H[M]
low value

Figure 2: The overview of our approach, taking "destroyed" as the candidate trigger. Our approach includes two complementary training objectives — an intra-sentence context-selective discrimination learning (left) and an inter-sentence contextualized similarity learning (right).

3.2 Context-Selective Discriminative Learning

Context-selective discriminative learning aims to predict the event label for the masked candidate trigger, by selectively attend to contexts. In our method, we first compute an (unsupervised) attention vector: $\boldsymbol{\alpha}_u$=softmax$(\boldsymbol{h}_{[\text{MASK}]}\boldsymbol{W}_a\boldsymbol{H}_S^{\mathsf{T}}) \in \mathcal{R}^{N+2}$, using $\boldsymbol{h}_{[\text{MASK}]}$, the representation of the masked candidate trigger as query vector (Bahdanau et al., 2014). $\boldsymbol{W}_a \in \mathcal{R}^{d\times d}$ is an attention matrix. Then we conduct a weighted summation computation over $\boldsymbol{H}_S$ using $\boldsymbol{\alpha}_u$ as the weight vector and compute a feature vector for the masked candidate trigger, denoted by $\boldsymbol{F}_{[\text{MS}]}$. Finally, $\boldsymbol{F}_{[\text{MS}]}$ is used for event label prediction by computing an output vector containing the probability of different event labels:

$$\boldsymbol{o}_{[\text{MS}]} = \boldsymbol{W}_m\boldsymbol{F}_{[\text{MS}]} + \boldsymbol{b}_m \qquad (1)$$

where $\boldsymbol{W}_m$ and $\boldsymbol{b}_m$ are model parameters. The predicted event label corresponds to the index having the highest value in $\boldsymbol{o}_{[\text{MS}]}$.

Considering that unsupervised attention may not always learn a good pattern (Wiegreffe and Pinter, 2019), we devise a "trial-and-error" approach to guide the learning. Specifically, at the training time, we also generate random context mask[3] and normalize it as a weight vector $\boldsymbol{\alpha}_r$. Our intuition is, if $\boldsymbol{\alpha}_r$ leads to a better result than using $\boldsymbol{\alpha}_u$, it might be a better selective pattern for our model to learn. Note there are cases where the predicted event labels are the same for $\boldsymbol{\alpha}_r$ and $\boldsymbol{\alpha}_u$, and here we introduce model uncertainty (Gal and Ghahramani, 2016) to evaluate whether the result is improved. Specifically, we compute the model uncertainty by making predictions many times but with dropout layers being activated, and the model uncertainty empirically equals to the prediction variance. When we note a reduced model uncertainty, we consider $\boldsymbol{\alpha}_u$ improves the result and we then encourage $\boldsymbol{\alpha}_u$ to approach $\boldsymbol{\alpha}_u$, under a guidance of mean square error (MSE) loss. Therefore, the overall loss function of context-selective discriminative learning is:

$$L_D = -\sum_t \log \boldsymbol{o}_{[\text{MS}]}[y_{(t)}] + \delta_{\alpha_u,\alpha_r}\text{MSE}(\boldsymbol{\alpha}_u, \boldsymbol{\alpha}_r) \qquad (2)$$

where t ranges over each token in the training set; $y_{(t)}$ is t's ground-truth event label; x$[j]$ denotes the jth element of x; $\delta_{\alpha_u,\alpha_r}$ takes a value of 1 if $\boldsymbol{\alpha}_r$ improves the result (regarding model uncertainty), and 0 otherwise.

3.3 Contextualized Similarity Learning

The philosophy of contextualized similarity learning is that "events of the same types may have similar contexts", derived from the distributional hypothesis of words (Harris, 1954). We enforce this assumption in our model by taking in pairs of mask-containing sentences as input, and have an objective to encourage their representations to be similar if they express the same type of events.

Let the learned feature vector of two (masked) candidate event triggers t_1 and t_2 be $\boldsymbol{F}_{\text{t}_1\to[\text{MS}]}$ and $\boldsymbol{F}_{\text{t}_2\to[\text{MS}]}$, and their event labels be y_1 and y_2. We define the similarity of $\boldsymbol{F}_{\text{t}_1\to[\text{MS}]}$ and $\boldsymbol{F}_{\text{t}_2\to[\text{MS}]}$ as:

$$\text{sim}_{t_1,t_2} = \frac{1}{1+\exp(\boldsymbol{F}_{\text{t}_1\to[\text{MS}]}\boldsymbol{F}_{\text{t}_2\to[\text{MS}]}^{\mathsf{T}})} \qquad (3)$$

[3] For example, a random mask might be [1, 1, 0, 1, ...], where 0 means that the third word is masked.

Based on this similarity measurement, we devise the following loss to encourage triggers of same types to have larger similarity score:

$$L_S = \sum_{t_1,t_2} - \delta_{y_1,y_2} \log(\text{sim}_{t_1,t_2}) + \quad (4)$$

$$(1 - \delta_{y_1,y_2}) \log(1 - \text{sim}_{t_1,t_2}) \quad (5)$$

where δ_{y_1,y_2} is the Kronecker function that takes 1 is y_1 and y_2 are same, and 0 otherwise. We do not consider cases where both of y_1 and y_2 are NIL.

3.4 Attentive Feature Fusion

Using only context-specific features for prediction may lead to sub-optimal performance. The attentive representation fusion is devised to balance the context-specific features and full contexts features, to make the reasoning more comprehensive.

Learning Full Context Features. The full context feature of a candidate trigger is learned in a similar way as in context-selective discriminative learning, but the candidate trigger is not masked and the context-selective attention is not performed. Note if we adopt a BERT-based full context feature learning, we can share the BERT encoder for full context feature learning and context-specific feature learning, and in this way, we do not need to double the model parameters. The impact of using other architectures for full context feature fusion is studied in § 6.1.

The Attentive Sentinel. The attentive sentinel aims to learn a trade-off between the context-specific feature $\boldsymbol{F}_{[MS]}$ and full context feature, denoted by $\boldsymbol{F}_{[FCT]}$ for a candidate trigger. Specifically, we first compute an attention weight via:

$$g = \sigma(\boldsymbol{W}_g[\boldsymbol{F}_{[MS]} \oplus \boldsymbol{F}_{[FCT]}] + b_g) \quad (6)$$

where $\boldsymbol{W}_g$ and $\boldsymbol{b}_g$ are model parameters. Then, using this weight, we compute a weighted summation of $\boldsymbol{F}_{[MS]}$ and $\boldsymbol{F}_{[FCT]}$ to compute the final feature of the candidate trigger:

$$\boldsymbol{F}_{com} = g\boldsymbol{F}_{[MS]} + (1 - g)\boldsymbol{F}_{[FCT]} \quad (7)$$

This attention mechanism enable us to learn a dynamically combination of the two features to make the final prediction.

3.5 Training and Optimization

Finally, in our full approach we take $\boldsymbol{F}_{com}$ as the input and conduct an event label classification via:

$$\boldsymbol{o}_{Final} = \boldsymbol{W}_f \boldsymbol{F}_{com} + \boldsymbol{b}_f \quad (8)$$

where $\boldsymbol{o}_{Final}$ contains probabilities of different event labels, and the predicted event label corresponds to the element have a maximal value; $\boldsymbol{W}_f$ and $\boldsymbol{b}_f$ are model parameters. A cross-entropy loss is adopted to train our full model, which is:

$$L_F = -\sum_t \log \boldsymbol{o}^t_{Final}[y_{(t)}] \quad (9)$$

where symbols have similar meanings as in Eq (2). We conduct a leaning paradigm of pre-training followed by fine-tuning: we first pre-train our model using L_D and L_S; then we fine-tune our model using L_F. In the later stage, L_F and L_D is also considered to keep the context-specific feature discriminative enough for prediction. We adopt Adam (Kingma and Ba, 2015) to update model parameters.

4 Experimental Setups

Datasets and Evaluations. We take ACE 2005 and KBP 2017 as the benchmark datasets. For ACE 2005, we split the corpus as training/developing/testing sets as recommend in previous works (Li et al., 2013; Chen et al., 2015). For KBP 2017, we adopt the official evaluation settings for training and testing. For evaluations, we adopt Precision (P), Recall (R), and F1-score (F1) as evaluation metrics, same as previous works for a meaningful comparison. We use two-tailed Wilcoxon test for significant test, with a significance level p=0.05.

Implementation Details. Our model is implemented with BERT$_{\text{Large}}$, which has 24 layers, 1024 hidden units, and 16 heads, and is pre-trained on large text corpora. We tune hyper-parameters via grid search on the developing set. Finally, the learning rate is set as $1e^{-5}$ (from [$1e^{-5}$, $2e^{-5}$ to $1e^{-4}$]); the batch size is set as 10 (from [2, 5 to 10]). A negative sampling rate of 0.7 is adopt to tackle the unbalance of positive and negative examples (Chen et al., 2015). As in KBP 2017 one event trigger might express multiple event types simultaneously, we adapt the multi-label cross entropy loss to binary cross-entropy loss, and a threshold of 0.3 is used for prediction.

Baselines. The following models are used as baselines: 1) DNNED, which adopts a feed-forward neural network for the task — it completely ignores context information; 2) DMCNN (Chen et al., 2015) and 3) RNNED (Nguyen et al.,

Model	Pre.	Rec.	F1
DNNED	68.6	64.9	66.7
DMCNN (2015)	75.6	63.6	69.1
JRNN (2016)	66.0	73.0	69.3
JMEE (2018b)	**76.3**	71.3	73.7
Delta-Adv (2019)	**76.3**	71.9	74.0
$\mathbf{M}_{\text{BERT}}$	74.1	73.1	73.6
$\mathbf{M}_{\text{FULL}}$	75.2	**74.4**	**74.8***
$\mathbf{M}_{\text{MASK}}$	47.7	42.7	45.0
$\mathbf{M}_{\text{MASK}}$ *w/o sel*	46.2	40.2	43.0
$\mathbf{M}_{\text{MASK}}$ *w/o sim*	45.0	40.6	42.7

Table 1: Results on ACE 2005. Pre., Rec., and F1 indicate precision (%), recall (%), and F1-score(%) respectively. Bold indicates the best result. * denotes a significance test with p=0.05. *w/o sel* and *w/o sim* denotes excluding context-selective attention and contextualized similarity learning respectively.

Model	Pre.	Rec.	F1
Top 3 System	54.3	46.6	50.1
Top 2 System	52.2	48.7	50.4
Top 1 System	56.8	55.6	56.2
Delta-Adv (2019)	62.3	53.7	57.7
$\mathbf{M}_{\text{BERT}}$	57.9	54.2	56.0
$\mathbf{M}_{\text{FULL}}$	**59.4**	**56.9**	**58.1***
$\mathbf{M}_{\text{MASK}}$	33.5	40.3	36.6
$\mathbf{M}_{\text{MASK}}$ *w/o sel*	32.4	39.2	35.5
$\mathbf{M}_{\text{MASK}}$ *w/o sim*	30.1	39.6	34.2

Table 2: Results on KBP 2017. Pre., Rec., and F1 indicate precision (%), recall (%), and F1-score(%) respectively. Bold indicates the best result. * denotes a significance test with p=0.05. *w/o sel* and *w/o sim* denotes excluding context-selective attention and contextualized similarity learning respectively.

2016), two state-of-the-art ED models employing Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNNs) for the task; 4) JMEE (Liu et al., 2018b), a graph model employs syntax information for the task. 5) Delta-Adv (Lu et al., 2019), a model that can learn discriminative and generalization features for the task via adversarial learning. 6) M_{BERT}, a model adopt BERT representations for ED. For KBP 2017, we also select the top 3 systems reported in the official evaluation as baselines. Among our models, the full model is denoted as $\mathbf{M}_{\text{FULL}}$; the model reasoning over the masked trigger is denoted as $\mathbf{M}_{\text{MASK}}$.

5 Experimental Results

5.1 Overall Performance

Table 1 and Table 2 show results of different models on ACE 2005 and KBP 2017 (we report 5-run average performance). From the results: 1) Our full approach $\mathbf{M}_{\text{FULL}}$ achieves the best performance, which outperforms all the baseline systems with a margin (+0.8% on F1 on ACE 2005 and +1.9% on KBP 2017). This demonstrates the effectiveness of our approach. Moreover, $\mathbf{M}_{\text{FULL}}$ consistently outperforms $\mathbf{M}_{\text{BERT}}$, which implies that the improvements do not simply come from introducing BERT representations for ED. 2) $\mathbf{M}_{\text{FULL}}$ also achieves the highest recall value, and this means that it can identify more positive examples than baselines, which may imply its ability in handling difficult cases that fail baselines. 3) Both of context-selective attention and contextualized similarity learning respectively can improve the performance, but the latter is more important — with out it, a model suffers from a drop of 2.3% in ACE 2005 and 2.4% in KBP 2017.

Another interesting finding is obtained by comparing DNNED with $\mathbf{M}_{\text{MASK}}$, which adopt only lexical or context information for the task. We conclude lexical information is much more important than context information in the standard evaluation. While, learning only such reasoning shortcuts may lead to poor robustness as shown in the following.

5.2 Robustness Probing

We conduct robustness probing regarding defending against adversarial attacks, unseen predicates, and tackling ambiguity cases. To maintain tractability, in the following experiments, we take model achieving best performance on the development set for testing, instead of adopting 5-run average as in previous evaluation. Moreover, to simplicity analysis, our experiments are mostly conducted on ACE 2005.

5.2.1 Defending Against Adversarial Attacks

In adversarial attacks, we adopt list-based method (Alzantot et al., 2018) to generate adversarial examples. Specifically, for a word, we first find its semantically similar words based on GloVe embeddings (Pennington et al., 2014), and then we replace the original word with each word and evaluate the new sentence with a GPT language model (Radford et al., 2019). We take the new sentence with the largest score as adversarial example. Some cases in

MODEL	ORG	ADT	ADC	ΔF1
DNNED	66.7	18.8	16.6	-47.1/50.1
DMCNN	69.0	20.1	19.2	-48.9/49.8
JRNN	69.5	19.3	18.9	-50.2/50.6
Delta-Adv	71.8	20.4	19.6	-51.4/52.2
$\mathbf{M}_{\text{BERT}}$	74.2	36.1	33.2	-38.1/41.0
$\mathbf{M}_{\text{FULL}}$	**76.0**	**47.9**	**43.3**	-28.1/32.7
$\mathbf{M}_{\text{MASK}}$	45.0	45.0	39.1	**-0/5.9**

Table 3: F1-score (%) of defending adversarial attacks. ORG indicates performance on the original testset. ADT and ADC indicate two types of adversarial attacks. ΔF1 indicates the performance gap.

MODEL	SEEN	UNSEEN	ΔF1
DNNED	74.9	14.2	-60.7
DMCNN (2015)	75.9	17.2	-58.7
JRNN (2016)	74.4	16.6	-57.8
Delta-Adv (2019)	75.1	17.8	-57.3
$\mathbf{M}_{\text{BERT}}$	75.6	25.2	-50.4
$\mathbf{M}_{\text{FULL}}$	**78.2**	**47.6**	-30.6
$\mathbf{M}_{\text{MASK}}$	58.1	31.1	**-27.0**

Table 4: F1 score (%) of exploring unseen predicts. SEEN indicates testing on the seen set, and UNSEEN indicates testing the the unseen set. ΔF1 indicates the performance gap.

MODEL	LA	HA	ΔF1
DNNED	70.6	50.4	-20.2
DMCNN (2015)	72.7	55.2	-17.5
JRNN (2016)	71.0	49.5	-21.5
Delta-Adv (2019)	72.2	52.1	-20.1
$\mathbf{M}_{\text{BERT}}$	73.5	60.3	-13.2
$\mathbf{M}_{\text{FULL}}$	75.6	63.4	-12.2
$\mathbf{M}_{\text{MASK}}$	49.7	50.7	-

Table 5: F1 score (%) of tackling ambiguity cases. LA indicates low-ambiguity cases; HA indicats high-ambiguity cases.

our approach are: 1) People in connection with the **killings** (→ **massacres**) that [...], 2) Anno-Marie **sued** (→ **alleged**) Crichton for divorce [...]. We perform two types of attack: ADT, attacking trigger words only; and ADC, attacking trigger words and context words.

From the results in Table 3, previous methods suffer from a severe drop (>47.1%/49.8%) in F1 facing adversarial attacks. By comparison, our full approach achieves the best performance — 47.9% and 43.3% regarding ADT and ADC respectively. $\mathbf{M}_{\text{MASK}}$ ranks secondly and demonstrates the smallest performance gap regarding adversarial attack — ADT even does not affect its performance as it does not rely on lexical information of trigger for prediction.

5.2.2 Exploring Unseen Predicates

The original testset may not be a good testbed for exploring unseen predicates, as it is highly biased (unseen cases only account for 8.1%). We adopt a new setting in exploring unseen predicates: we first divide the whole ACE corpus as C1 and C2 with a ratio of 1:2 randomly, and C1 is used for training/developing. Then, for each sentence in C2, we put it into a SEEN or UNSEEN set based on whether it contains a trigger that is in C1 or not (for sentence that does not have event triggers, we randomly put it into the SEEN or UNSEEN set). Finally, we end up with a SEEN set with a size of 2, 896, and an UNSEEN set with a size of 1, 409.

Table 4 show the results of different models. We note previous methods behave poorly on the UNSEEN set and demonstrate a large performance gap (>50.4%) in handle SEEN and UNSEEN. By contrast, our full approach achieves the best performance on SEEN (78.2%) and UNSEEN (47.6%), with a relatively small gap (30.6%). Moreover, $\mathbf{M}_{\text{MASK}}$ ranks secondly on the UNSEEN set, outperforming all other baselines including $\mathbf{M}_{\text{BERT}}$.

5.2.3 Tackling Ambiguity Cases

Regarding tackling ambiguity cases, we first define the ambiguity of a word as the entropy of its word-type distribution. We then sort all sentences based on their averaged word ambiguity. For example, a high-ambiguity sentence is “There was no shots fired”, where “shots” can trigger *Attack*, *Die*, *Execute*, and *NIL* and “fired” can trigger *Attack*, *End-Position*, and *NIL*. We select 500 sentences with the highest ambiguity to construct a HA set; 500 sentences with the lowest ambiguity to construct a LA set (each of the sentence should contain at least one event trigger).

From the results shown in Table 5, previous ED systems (except M_{BERT}) have a relatively large performance gap in tackling low-ambiguity and high-ambiguity cases. By contrast, our full approach achieves the best performance with a small gap. Interestingly, $\mathbf{M}_{\text{BERT}}$ demonstrates a rather good

EXAMPLE	GOLDEN	$\mathbf{M}_{\text{BERT}}$	$\mathbf{M}_{\text{MASK}}$
a) The EU is set to **release** 20 million euros ...	Transfer-Money	Release-Parole ✗	Transfer-Money ✓
b) ... British budget cinema chain **launched** by the founder ...	Start-Org	Transport ✗	Start-Org ✓
c) missiles capable of **reaching** Israel and possibly weapons	Attack	NIL ✗	Attack ✓
d) ..., it **admits** troops into the country for Iraq conflict ...	Transport	NIL ✗	Transport ✓
e) She failed to **become** a deputy in the parliament ...	Start-Position	Start-Position ✓	Elect ✗
f) No convict has ever been **executed** in the country.	Execute	Execute ✓	Arrest-Jail ✗
g) Campbell, 55, was **pulled** over Jan. 10 after police ...	NIL	NIL	Arrest-Jail **?**
h) In an **address** Saturday, Information Minister [...]	NIL	NIL	Contact-Meet **?**
i) That [...] detailed monthly **outlays** of some 51, 000 ...	NIL	NIL	Transfer-Money **?**

Table 6: Examples exploring the predictive bias of $\mathbf{M}_{\text{BERT}}$ and $\mathbf{M}_{\text{MASK}}$. Event triggers are in bold. GOLDEN denotes the ground-truth labels.

MODEL	ORG	+$\mathbf{M}_{\text{MASK}}$	ΔF1
DNNED	66.7	71.4	**+4.7**
DMCNN (2015)	69.0	72.3	+3.3
JRNN (2016)	69.5	72.9	+3.4
JMEE (2016)	71.8	72.9	+1.1

Table 7: F1 score (%) of integrating M_{BERT} with existing ED models.

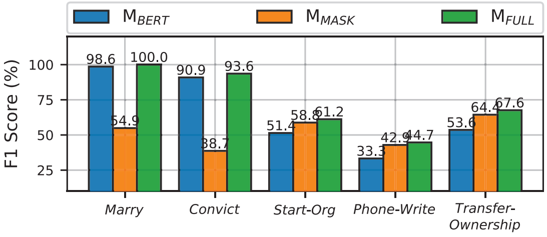

Figure 3: Performance of $\mathbf{M}_{\text{BERT}}$, $\mathbf{M}_{\text{MASK}}$, and $\mathbf{M}_{\text{FULL}}$ on different event types.

performance in tackling ambiguity cases, which may benefit from its ability in modeling contexts by pre-training on large corpus. We also note $\mathbf{M}_{\text{MASK}}$ show comparable performance in tackling low- and high-ambiguity cases.

6 Further Discussion

6.1 Predictive Bias Probing

We first explore the integration of M_{MASK} with existing ED models learning full context features. From the results in Table 7, M_{MASK} has a complementary effect with existing ED systems and boosts performance. The gain on DNNED is the most salient, as DNNED only uses trigger information but context information for reasoning, which is the opposite of M_{MASK}. Additionally, we compare performance of $\mathbf{M}_{\text{BERT}}$, $\mathbf{M}_{\text{MASK}}$, and $\mathbf{M}_{\text{FULL}}$ on different event types in Figure 3. From the results $\mathbf{M}_{\text{BERT}}$ performs better on types having relatively fewer expressions such as *Marry* and *Convict*, but worse on types having diverse expressions such as *Start-ORG*, *Phone-Write*, and *Transfer-Ownership*. $\mathbf{M}_{\text{MASK}}$ is just the opposite. $\mathbf{M}_{\text{FULL}}$ can take advantages of feature fusion from $\mathbf{M}_{\text{BERT}}$ and $\mathbf{M}_{\text{MASK}}$, yielding the best performance.

6.2 Case Study

We conduct case study to explore the outputs of $\mathbf{M}_{\text{BERT}}$ and our model $\mathbf{M}_{\text{MASK}}$, and the representative and interesting cases are shown in Table 6. From the results, in a) and b), $\mathbf{M}_{\text{BERT}}$ makes wrong predictions, which may due to the prevalent of the pattern **release** → *Transfer-Money* (100%) and **launched** → Transport (78.5%) in the training set. $\mathbf{M}_{\text{BERT}}$ also misses c) and d), as the detection of **reaching** and **admits** is completely depended on contexts. By contrast, $\mathbf{M}_{\text{MASK}}$ correctly identify all of them.

More interesting cases are shown in the second part of Table 6. We note our model $\mathbf{M}_{\text{MASK}}$ makes wrong predictions in e) and f). This makes sense, as $\mathbf{M}_{\text{MASK}}$ does not aware trigger lexical information — even human may wrongly predict an Arrest-Jail event considering "convict has ever been [MASK] in [...]". Example g), h) and i) are worth further discussion. From our opinion, $\mathbf{M}_{\text{MASK}}$ assigns an *Arrest-Jail* event to **pull** in g), and a *Contact-Meet* event to **address** in h), which are quite reasonable. But these cases are not labeled in the golden annotations, which may be missed by the ACE annotators. This also implies the challenging of the ED task.

7 Conclusion and Future Work

This paper focuses on the robustness of ED. We highlight three stark cases showing the brittleness of existing ED models. Then we propose a new approach called context-selective masking generalization shedding lights on robustifying an ED model. In future, we would like to extend our method to other tasks where exploiting context information is crucial, such as named entity recognition and relation extraction.

Acknowledgments

This work is supported by the Natural Key R&D Program of China (No.2018YFB1005100), the National Natural Science Foundation of China (No.61533018, No.61976211, No.61806201) and the Key Research Program of the Chinese Academy of Sciences (Grant NO. ZDBS-SSW-JSC006). This work is also supported by Huawei Technologies Co., Ltd. , Beijing Academy of Artificial Intelligence (BAAI2019QN0301) and independent research project 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*.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *EMNLP*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. Cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *CoRR*, abs/1711.02173.

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D. Manning. 2014. Modeling biological processes for reading comprehension. In *EMNLP*.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On adversarial examples for character-level neural machine translation. In *COLING*.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. 2016. A language-independent neural network for event detection. In *ACL*.

Elena Filatova and Vasileios Hatzivassiloglou. 2004. Event-based extractive summarization. In *Text Summarization Branches Out*, pages 104–111.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

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.

Yu-Lun Hsieh, Minhao Cheng, Da-Cheng Juan, Wei Wei, Wen-Lian Hsu, and Cho-Jui Hsieh. 2019. On the robustness of self-attentive models. In *ACL*.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *ACL*.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *ACL*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *EMNLP*.

Yichen Jiang and Mohit Bansal. 2019. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA. In *ACL*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *EMNLP*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *ACL*.

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

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany. 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. 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 6754–6761. AAAI Press.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2018a. Event detection via gated multilingual attention mechanism. In *AAAI*.

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.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *ACL*.

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

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.

Teruko Mitamura, Zhengzhong Liu, and Eduard H Hovy. 2017. Events detection, coreference and sequencing: What's next? overview of the tac kbp 2017 event track. In *TAC*.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the model understand the question? In *ACL*.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *NAACL*.

Nicolas Papernot, Patrick D. McDaniel, Ananthram Swami, and Richard E. Harang. 2016. Crafting adversarial input sequences for recurrent neural networks.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

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

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *ICLR*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. 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 11–20, Hong Kong, China. Association for Computational Linguistics.

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