DESED: Dialogue-based Explanation for Sentence-level Event Detection

Yinyi Wei[1*†], **Shuaipeng Liu**[2*‡], **Jianwei Lv**[2],
Xiangyu Xi[2], **Hailei Yan**[2], **Wei Ye**[3‡], **Tong Mo**[1], **Fan Yang**[2], **Guanglu Wan**[2]
[1] Peking University
[2] Meituan Group, Beijing, China
[3] National Engineering Research Center for Software Engineering, Peking University
wyyy@pku.edu.cn, liushuaipeng@meituan.com, wye@pku.edu.cn

Abstract

Many recent sentence-level event detection efforts focus on enriching sentence semantics, e.g., via multi-task or prompt-based learning. Despite the promising performance, these methods commonly depend on label-extensive manual annotations or require domain expertise to design sophisticated templates and rules. This paper proposes a new paradigm, named dialogue-based explanation, to enhance sentence semantics for event detection. By saying dialogue-based explanation of an event, we mean explaining it through a consistent information-intensive dialogue, with the original event description as the start utterance. We propose three simple dialogue generation methods, whose outputs are then fed into a hybrid attention mechanism to characterize the complementary event semantics. Extensive experimental results on two event detection datasets verify the effectiveness of our method and suggest promising research opportunities in the dialogue-based explanation paradigm.

1 Introduction

Event detection (ED) is a crucial task in information extraction, which aims to identify event triggers (words or phrases that indicate events) and classify triggers into predefined event types. For example, we can identify the trigger *weddings* and classify it into *Marry* event type from the text "Giuliani regularly officiated at *weddings* while in office". Sentence-level event detection plays a dominant role in event detection and is significant for various downstream NLP tasks.

However, it is usually challenging to accurately detect events in a single sentence due to the limited information. Therefore, most prior methods on sentence-level event detection make improvements by enhancing sentence semantics, being divided into two categories. The first category mainly involves leveraging other information extraction tasks (e.g., named entity recognition and relation extraction) via multi-task learning (Wadden et al., 2019; Lin et al., 2020; Van Nguyen et al., 2021). However, these efforts highly depend on task-specific annotation, costing a vast amount of human effort. The other popular line of research exploits pretrained language models (PLMs), e.g., via prompt-based learning(Gao et al., 2021; Lee et al., 2021; Li et al., 2022; Hsu et al., 2022). MRC-based methods, which treat a task as a Machine Reading Comprehension task (Liu et al., 2020; Li et al., 2020; Du and Cardie, 2020), can also be regarded as a weaker version of prompt-based learning. One common bottleneck among these methods lies in their reliance on domain expertise and human efforts to devise sophisticated templates and rules.

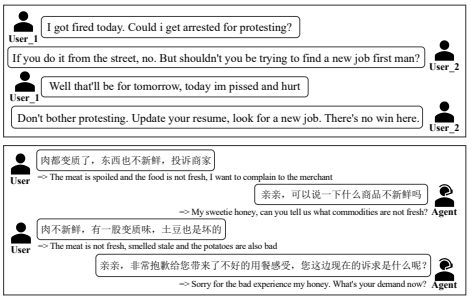

Figure 1: Two examples of dialogues from Reddit and FOSAED, respectively.

To enhance sentence semantics more effectively and efficiently, this work proposes to use generative models to generate contextual information for a sentence in the form of a dialogue, which consists of multiple utterances between different roles on a particular topic.

As two motivation examples, Figure 1 shows two real-world dialogues. In the utterance from User_1 in the first example, models are easily induced by *arrested* and *protesting* thus identifying *fired* as an *attack* event, but the subsequent utterances serve as

*Equal Contribution.
†Work was done when Yinyi Wei interned at Meituan.
‡Corresponding Author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 2483–2493
October 12–17, 2022.

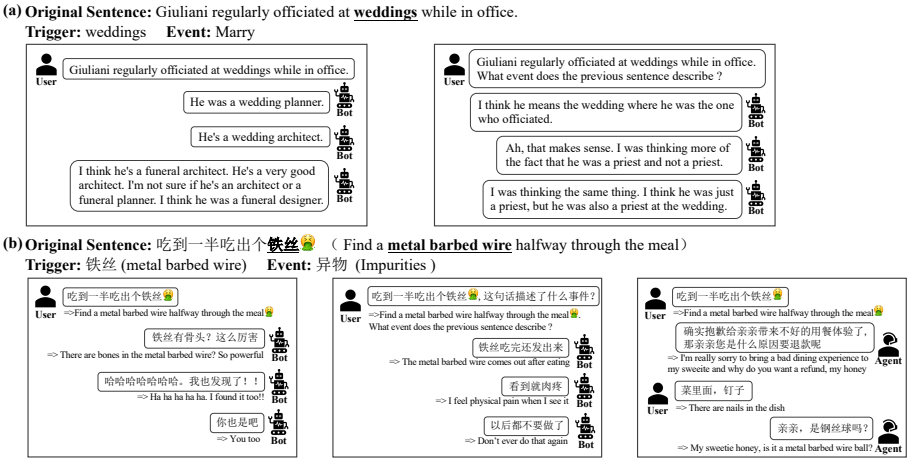

Figure 2: Examples of dialogue generation for a specific sentence with three methods: (1) Direct generation; (2) Generation with a prompt; (3) Further training and generation. Figure (a) shows the dialogue generation using method (1)(2) on ACE05-E$^+$. Figure (b) shows the dialogue generation using method (1)(2)(3) on FOSAED-R.

an explanation that *fired* is an *End-Position* event. In the second example, the dialogue provides clues about a natural association among multiple events, including physical feelings of user, food quality and complaints about the restaurant. Based on these two examples, we conjecture two main merits of dialogues over plain narrative texts in terms of enriching event context. On the one hand, a dialogue is more consistent with the original sentence (see Section 4.5 and 4.6). On the other hand, each utterance is an independent semantic unit requiring no additional segmentation, which is non-trivial for a plain text generated, e.g., by GPT-2. And more importantly, the interaction between these utterances provides room for refining the dialogue-based context. In this paper, we refer the generated dialogue for an event description to dialogue-based explanation and call our method **DESED**: **D**ialogue-base **E**xplanation for **S**entence-level **E**vent **D**etection.

In order to generate semantically rich dialogue-based explanation, we propose three methods based on pretrained dialogue GPTs (Radford et al., 2018, 2019): (1) direct generation on the original sentence; (2) generation with a prompt on the original sentence; (3) generation after further training on dialogue data in the same domain. The three methods are illustrated in Figure 2. Note that prompts we use are quite simple, and identical prompts can be used in our dialogue generation for different events and datasets. In contrast, the aforementioned prompt-based methods require redesigning templates and prompts, demanding expertise across different domains.

To exploit the information of generated dialogues, we then propose three methods: (1) token-level attention with the self-attention mechanism of PLMs; (2) utterance-level attention with an utterance gate; (3) hybrid attention combining the both. We conduct experiments on ACE2005 and another event detection dataset based on real-world data curated by ourselves. Experimentally, our method achieves competitive performance than previous multi-task and prompt-based works.

Our main contributions include:

- We propose dialogue-based explanation, a novel paradigm to enrich sentence semantics for event detection by generating a consistent dialogue on specific events.
- We propose three conceptually simple methods to generate dialogue-based explanation and design hybrid (token-level and utterance-level) attention mechanisms that demonstrate competitive results on two datasets.
- Our experiments reveal that compared with plain narrative contexts, dialogues are more consistent with original sentences and contain richer contextual knowledge for event detection, and appropriate prompts or dialogue data in a specific domain can guide pretrained models to generate better event-centric dialogues.

2 Related Work

2.1 Sentence-level Event Detection

To identify a trigger and classify the trigger into an event type from a sentence, traditional feature-based methods rely heavily on manually designing

features (Ahn, 2006; McClosky et al., 2011). With the development of deep learning, neural networks have been widely used in event detection. The most common usage for neural networks is token classification, which encodes and classifies each token with various neural methods (Chen et al., 2015; Nguyen et al., 2016; Sha et al., 2018). Furthermore, graph based (Liu et al., 2018; Yan et al., 2019), multi-task (Wadden et al., 2019; Lin et al., 2020; Van Nguyen et al., 2021; Lu et al., 2022), MRC-based (Liu et al., 2020; Li et al., 2020; Du and Cardie, 2020), Seq2Seq-based(Sequence-to-Sequence-based) (Lu et al., 2021; Hsu et al., 2022; Paolini et al., 2021) methods have also been introduced to sentence-level event detection.

2.2 Prompt-based Learning

Prompt-based learning aims to stimulate the knowledge of PLMs to serve downstream tasks (Schick and Schütze, 2021). Unidirectional language models (e.g. GPTs (Radford et al., 2018, 2019)), bidirectional language models (e.g. BERT (Kenton and Toutanova, 2019)) and hybrid language models (e.g. BART (Lewis et al., 2020)) can all be used as backbones. By retrieving similar instances in the training set or adding manual definitions of labels (Gao et al., 2021; Lee et al., 2021; Kumar and Talukdar, 2021), or by converting information extraction tasks to slot-filling tasks (Lu et al., 2021; Hsu et al., 2022; Li et al., 2022), prompt-based learning enables PLMs to have priori knowledge of a task, thus contributing to the final performance.

2.3 Generation-based Dialogue System

Generation-based dialogue system can generate a great diversity of responses which are not limited to the existing corpus (Chen et al., 2017). By making use of GPTs (Radford et al., 2018, 2019) and large amount of dialogue data, generation-based models can achieve excellent results on different languages (Zhang et al., 2020; Wang et al., 2020).

3 Methodology

In this section, we present our dialogue-based explanation for sentence-level event deteiction.

3.1 Task Description

In this paper, we formulate sentence-level event detection as a sequence labelling task using *BIO* tagging format. Given a trigger which evokes an event *EventType*. Each token is tagged as *B-EventType*, *I-EventType* or *O*, indicating the token is at the beginning, inside or outside of the trigger tokens.

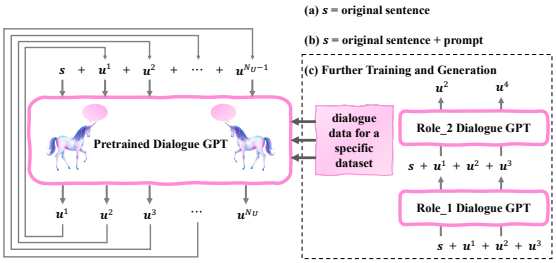

Figure 3: Illustration of dialogue generation methods and an example of dialogue generation with further training on two roles.

Formally, denote $\mathcal{S}$, $\mathcal{Y}$, $\mathcal{M}$ as instance set, label set and bidirectional language model. For a sentence instance $\boldsymbol{s} \in \mathcal{S}$, $\boldsymbol{s} = (s_0, s_1, \ldots, s_{N_s-1})$. In the general setting, representation $\boldsymbol{h} = \mathcal{M}(\boldsymbol{s})$, $\boldsymbol{h} \in \mathbb{R}^{N_s \times D}$, where D is the hidden size of $\mathcal{M}$. When using *BIO* tagging format, the set of all tags is $\mathcal{E}$, the total number of $\mathcal{E}$ is $|\mathcal{E}| = 2 \times |\mathcal{Y}| + 1$. To conduct sequence labelling, a weight matrix $\boldsymbol{W} \in \mathbb{R}^{D \times |\mathcal{E}|}$ and a bias term $\boldsymbol{b} \in \mathbb{R}^{|\mathcal{E}|}$ are introduced to classify each token representation into a tag in $\mathcal{E}$. The classification logits $\boldsymbol{p} = \boldsymbol{hW} + \boldsymbol{b}$, $\boldsymbol{p} \in \mathbb{R}^{N_s \times |\mathcal{E}|}$. The final labelling results $\boldsymbol{e} = \text{argmax}(\boldsymbol{p})$, $\boldsymbol{e} \in \mathbb{R}^{N_s}$,where e_i is the tag of s_i. The optimization objective is set to a cross entropy loss between classification logits $\boldsymbol{p}$ and golden tagging.

3.2 Dialogue Generation

A pretrained dialogue generation model $\mathcal{G}$ is used to generate dialogues. The overview of dialogue generation is shown in Figure 3.

3.2.1 Direct Generation

Given a sentence instance $\boldsymbol{s} = (s_0, s_1, \ldots, s_{N_s-1})$, the goal is to generate N_U utterances. $\boldsymbol{s}$ is firstly fed into $\mathcal{G}$ to obtain an utterance $\boldsymbol{u}^1$, $\boldsymbol{u}^1 = \mathcal{G}(\boldsymbol{s})$. Then $\boldsymbol{s}$ and $\boldsymbol{u}^1$ are concatenated as dialogue history which is fed into $\mathcal{G}$ to get a new response utterance $\boldsymbol{u}^2$, $\boldsymbol{u}^2 = \mathcal{G}(\boldsymbol{s} + \boldsymbol{u}^1)$. Circulating repeatedly, until $\boldsymbol{u}^{N_U}$ is obtained, $\boldsymbol{u}^{N_U} = \mathcal{G}(\boldsymbol{s} + \boldsymbol{u}^1 + \cdots + \boldsymbol{u}^{N_U-1})$.

3.2.2 Generation with a Prompt

To make the generated dialogue more focused on a particular topic, we propose to adding a straightforward prompt at the end of the original sentence, (e.g. *What event does the previous sentence describe?*), which means $\boldsymbol{s} = \boldsymbol{s} + \text{prompt}$. The procedure described in 3.2.1 is then repeated until N_U utterances are obtained.

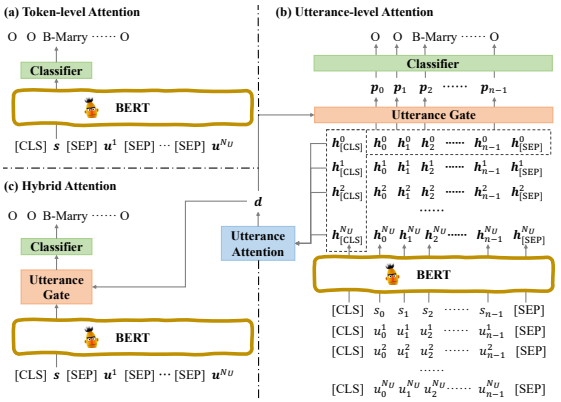

Figure 4: Different attention mechanisms of exploiting dialogue information. Figure (a) illustrates the token-level attention; Figure (b) illustrates the utterance-level attention; Figure (c) illustrates the hybrid attention.

3.2.3 Further Training and Generation

When dialogue data is provided for a dataset, further training can be carried out based on this data.

For the dialogue data with k roles, k different dialogue models are trained with role-specific responses in order to model the characteristics of different roles. When inferring, k different dialogue models are used alternatively to generate utterances from different roles. An example of dialogue generation on two roles is shown on the right of Figure 3.

3.3 Exploitation of Dialogue Information

We exploit generated dialogue information through different attention mechanisms based on sequence labelling. The overview is illustrated in Figure 4.

3.3.1 Token-level Attention

By encoding the concatenation of the original sentence and generated utterances simultaneously with a bidirectional language model $\mathcal{M}$, we can take advantage of the self-attention mechanism and the ability to capture long-range dependencies in $\mathcal{M}$.

Given a sentence instance $\boldsymbol{s}$ and generated utterances $\boldsymbol{u}^1, \ldots \boldsymbol{u}^{N_U}$, we use the separator token of $\mathcal{M}$ (e.g. [SEP] for BERT) to concatenate the original sentence and all utterances. Thus the combined input $\boldsymbol{c} = \boldsymbol{s}$ [SEP] $\boldsymbol{u}^1$ [SEP] $\ldots$ [SEP] $\boldsymbol{u}^{N_U}$. After obtaining contextual representations by feeding $\boldsymbol{c}$ into $\mathcal{M}$, the token representations corresponding to $\boldsymbol{s}$ are classified into specific tags by a classifier.

3.3.2 Utterance-level Attention

Due to the uncertainty of $\mathcal{G}$, generated utterances may be disorganized and rambling. Directly combining and applying self-attention mechanism may introduce noise to the representation of the original sentence. We therefore propose to use an utterance attention mechanism and an utterance gate to integrate dialogue information into the representation of the original sentence.

Given a sentence instance $\boldsymbol{s}$ and generated utterance $\boldsymbol{u}^1, \ldots, \boldsymbol{u}^{N_U}$, assuming that the original sentence and all utterances are of length n. As shown in Figure 4(b), feeding them into $\mathcal{M}$, we can obtain representation $\boldsymbol{h} = (\boldsymbol{h}^0, \boldsymbol{h}^1, \ldots, \boldsymbol{h}^{N_U})$, where $\boldsymbol{h}^0$ is the representation of $\boldsymbol{s}$; $\boldsymbol{h}^j$, $j \geq 1$, is the representation of $\boldsymbol{u}^j$. For all $\boldsymbol{h}^i$, $i \geq 0$, $\boldsymbol{h}^i \in \mathbb{R}^D$.

An attention mechanism is applied to get a dialogue state $\boldsymbol{d}$ with the representation of [CLS] token $\boldsymbol{h}^i_{[\text{CLS}]}$ and learned attention weight α_i:

$$\boldsymbol{d} = \sum_{i=0}^{N_U} \alpha_i \boldsymbol{h}^i_{[\text{CLS}]},\ \boldsymbol{d} \in \mathbb{R}^D \tag{1}$$

$$\alpha_i = \frac{\exp(s_i)}{\sum_{j=0}^{N_U} \exp(s_j)} \tag{2}$$

$$s_i = \tanh\left(\boldsymbol{h}^0_{[\text{CLS}]} \cdot (\boldsymbol{W}_a \cdot (\boldsymbol{h}^i_{[\text{CLS}]})^T + \boldsymbol{b}_a)\right) \tag{3}$$

where $\boldsymbol{W}_a$ and $\boldsymbol{b}_a$ are the weight matrix and the bias term of a feed-forward neural network, s_i is the relevance score between the original sentence $\boldsymbol{s}$ and an generated utterance $\boldsymbol{u}^i$.

Knowing that $\boldsymbol{d}$ is the semantic abstraction of the whole dialogue, we further propose an utterance gate to fuse $\boldsymbol{d}$ into token representations of $\boldsymbol{s}$.

For the representation of the original sentence $\boldsymbol{h}^0 = (\boldsymbol{h}^0_0, \boldsymbol{h}^0_1, \ldots, \boldsymbol{h}^0_{n-1})$, the fused representation $\boldsymbol{p} = (\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{n-1})$ is computed as below:

$$\boldsymbol{p}_i = \boldsymbol{h}^0_i \parallel \boldsymbol{f}_i \tag{4}$$

$$\boldsymbol{f}_i = \theta_i \circ \boldsymbol{h}^0_i + (1 - \theta_i) \circ \boldsymbol{d} \tag{5}$$

$$\theta_i = \text{sigmoid}((\boldsymbol{h}^0_i \parallel \boldsymbol{d}) \cdot \boldsymbol{W}_g + b_g) \tag{6}$$

where $\parallel$ is the notation for the concatenation of two vectors, $\circ$ indicates scalar multiplication, $\boldsymbol{W}_g$ and b_g are the weight matrix and the bias term of a feed-forward neural network. θ can be seen as a dynamic threshold to determine how much dialogue information needs to be incorporated into token representations. A classifier is then applied on $\boldsymbol{p}$ to get the final tagging result.

3.3.3 Hybrid Attention

To cover different levels of attention, we propose to use attention mechanisms at both token-level and utterance-level. To get a representation $\boldsymbol{h}^c$ with token-level attention, combined sentence $\boldsymbol{c}$

Form	#Docs	#Sents
Labelled User Reviews	4,226	4,226
Unlabelled Conversations	7,155	309,295

Table 1: Statistics of FOSAED. We show the number of documents and sentences for different forms of data.

Dataset	Split	#Sents	#Events
ACE05-E$^+$	Train	19,216	4,419
	Dev	901	468
	Test	676	424
FOSAED-R	Train	3,380	3,893
	Dev	423	494
	Test	423	512

Table 2: Dataset statistics. We show the number of sentences and events for different splits.

is fed into $\mathcal{M}$: $\boldsymbol{h}^c = \mathcal{M}(\boldsymbol{c})$. Then the utterance attention mechanism and utterance gate are applied to compute the dialogue state $\boldsymbol{d}$ and fuse $\boldsymbol{d}$ into $\boldsymbol{h}^c$. Finally token classification is conducted on the fused representations corresponding to $\boldsymbol{s}$.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets and Evaluation Metrics

We evaluate on two event detection datasets, ACE2005 (Doddington et al., 2004) and FOSAED.

ACE2005, a collection of documents from a diversity of domains, is the most widely used dataset for event extraction. For data split and preprocessing, we follow Lin et al. (2020), which adds back pronouns and multi-token triggers. We use the English version which covers 8 event types and 33 event subtypes and refer to it as ACE05-E$^+$.

Aiming at evaluating DESED on a specific domain, we curate and propose a new dataset named FOSAED (Food Safety on User Reviews for Event Detection). FOSAED is a real-world Chinese event detection dataset, consisting of sentence-level user reviews (reviews posted by users about orders and restaurants) in the domain of food safety based on a leading e-commerce platform for food service. FOSAED focuses on 4 event types and 21 event subtypes. Each event type and event subtype correspond to a food safety issue (e.g. Abnormalities, Uncomfortable and Undercooked). To support further training, a number of unlabelled conversations are collected, which are in the same domain (i.e., food safety) as the user reviews. These conversations are dialogues between users and agents, and have two sources: text conversations (users communicate online with after-sale agents via text messages) and phone conversations (users communicate with after-sale agents via telephone). Statistics of FOSAED are shown in Table 1. We treat the conversations as the further training dialogue data and conduct event detection on the user reviews. The version is denoted as FOSAED-R.

Statistics of ACE05-E+ and FOSAED-R are shown in Table 2.

For evaluation, we use the same criteria in previous work (Li et al., 2013; Wadden et al., 2019; Lin et al., 2020) and report F1-scores in our experiments. **Trig-I**: A trigger is correctly identified if its offset match any of the gold triggers. **Trig-C**: The span of the trigger is correctly identified and its event type is also correctly classified.

4.1.2 Baselines

We compare DESED to baselines with multi-task learning and prompt-based learning. Specifically, we compare with: (1) **BILSTM+CRF**(Hochreiter and Schmidhuber, 1997; Lafferty et al., 2002), using a bi-directional long short-term memory network and a conditional random field layer; (2) **DMBERT**(Wang et al., 2019), using BERT and dynamic multi-pooling mechanism to assemble features; (3) **BERT**(Kenton and Toutanova, 2019), fine-tuning BERT for token classification; (4) **BERT_QA_TRIGGER**(Du and Cardie, 2020), converting event detection to a MRC task; (5) **OneIE**(Lin et al., 2020), a span-based model with multi-task learning; (6) **FourIE**(Van Nguyen et al., 2021), a span-based model using Graph Convolutional Networks with multi-task learning; (7) **Text2Event**(Lu et al., 2021), using a Seq2Seq model to generate a manually designed structure for each event; (8) **DEGREE**(Hsu et al., 2022), taking advantage of a Seq2Seq model with manually designed templates and prompts; (9) **PILED**(Li et al., 2022), using a prompt-based method to identify a event then adding event-specific demonstration to localize a trigger; (10) **TANL**(Paolini et al., 2021), treating multi-task as translation between augmented natural language and predicting structures with designed annotations; (11) **UIE**(Lu et al., 2022), using a unified text-to-structure generation with multi-task and prompt-based learning.

Category	Methods		ACE05-E$^+$		FOSAED-R	
			Trig-I	Trig-C	Trig-I	Trig-C
Basic	BiLSTM+CRF		72.9	69.3	71.5	70.8
	DMBERT		73.5	69.5	72.8	71.4
	BERT		73.4	70.5	73.6	71.5
MRC-based	BERT_QA_TRIGGER		74.6	71.5	72.9	71.8
Multi-task	OneIE*		75.6	72.8	-	-
	FourIE*		76.7	73.3	-	-
Prompt-based	Text2Event*		-	71.8	-	-
	DEGREE*		76.7	72.7	-	-
	PILED*		-	73.4	-	-
Multi-task and Prompt-based	TANL*		71.5	68.4	-	-
	UIE*		-	73.4	-	-
Dialogue-based Explanation	DESED	Direct Generation	76.2	72.3	**75.8**	74.3
		Generation with a Prompt	**76.9**	**73.5**	**75.8**	74.3
		Further Training	-	-	75.6	**74.4**

Table 3: Experimental results of sentence-level event detection on ACE05-E$^+$ and FOSAED-R (F1-score, %). The best results are in boldface. * indicates results cited from the original paper.

4.1.3 Implementation Details

For all experiments on sequence labelling, we select AdamW for optimization with a learning rate of 3e-5, weight decay of 5e-5, adam ϵ of 1e-8 and max gradient norm of 1.0. The max sequence length is set to the max token length in a batch, and the total max sequence length is set to 256 for ACE05-E$^+$ and 512 for FOSAED-R. We use a linear layer with a dropout rate of 0.3 for the classifier. Each model is trained for 10 epochs and choose the checkpoint with the best validation performance on the development set. For ACE05-E$^+$, we use a batch size of 4 and gradient accumulation step of 4, and BERT-large is applied as backbone. For FOSAED-R, we use a batch size of 4 and gradient accumulation step of 2, and BERT-base-Chinese is applied as backbone. We do all the experiments on NVIDIA Tesla V100. Our codes and datasets are released at https://github.com/Ydongd/DESED.

In order to generate grammatically correct and semantically rich dialogues, we use DialoGPT-large for ACE05-E$^+$ and CDial-GPT$_{\mathrm{LCCC-large}}$ for FOSAED-R as pretrained dialogue generation models. Four prompts are used to generate dialogues. We generate 1-5 utterances from an original sentence and report the best results. For further training on dialogue data, since there is no suitable and sufficient dialogue data in ACE05-E$^+$, we only conduct further training on FOSAED-R.

For unlabelled conversations in FOSAED, we first eliminate mechanical responses according to rules and merge consecutive utterances with the same role, then select the utterances with events (detected by a BERT model) and the next five responses from those utterances as the dialogue dataset which is used to train a user dialogue model and an agent dialogue model. For the user dialogue model, there is 36,395 dialogues in the training set and 4,678 dialogues in the development set; while for the agent dialogue model, there is 36,236 dialogues in the training set and 4,630 dialogues in the development set. When further training, we use a learning rate of 3e-5 and a max gradient norm of 1.0. We train the model for 10 epochs with 5000 warmup steps. The batch size is set to 8 and the gradient accumulation steps is set to 32, which is equivalent to a batch size of 256.

4.2 Main Results

From Table 3, we can see that DESED outperforms basic sequence labeling models (e.g., BiLSTM+CRF and DMBERT) as expected. Compared with the robust BERT token classification method, DESED also achieves improvements of 4.3% Trig-C F1 on ACE05-E$^+$ (73.5% v.s. 70.5%) and 4.1% Trig-C F1 on FOSAED-R (74.4% v.s. 71.5%). The superiority of DESED can also be easily observed by comparing it against a series of multi-task and prompt-based methods. These results prove the overall feasibility and effectiveness of our dialogue-based explanation paradigm.

On ACE05-E$^+$, generation with a prompt yields better results than direct generation. The possible

Generation	Att	ACE05-E$^+$		FOSAED-R	
		Trig-I	Trig-C	Trig-I	Trig-C
Direct	T	74.6	71.6	75.8	74.3
	U	74.9	71.8	75.0	73.4
	H	76.2	72.3	75.7	73.8
Prompt	T	75.2	72.3	75.1	73.7
	U	76.2	73.5	75.8	74.3
	H	76.9	73.3	74.3	72.9
Further	T	-	-	74.3	72.9
	U	-	-	74.9	73.5
	H	-	-	75.6	74.4

Table 4: Different attention mechanisms of DESED on ACE05-E$^+$ and FOSAED-R (F1-score, %). T, U and H denote token-level, utterance-level and hybrid attention mechanism respectively.

reason lies in that sentences in ACE05-E$^+$ are cut from documents, and many are unstructured, making it difficult to generate high-quality dialogues directly. While a clear and clarified prompt bridges the gap between unstructured sentences and generated utterances. On FOSAED-R, different methods produce similar results, as each user review is a complete and independent sentence.

4.3 Attention Mechanisms in DESED

Results of different attention mechanisms are shown in Table 4. Intuitively, more complex attention leads to better performance. However, this is not the case from the experimental results. There are two main reasons: firstly, generated dialogues have many noises and cannot simply be treated as standard contextual texts; secondly, there are differences in the training data for pretrained dialogue generation models. The English and Chinese datasets are constructed from Reddit comments and Weibo conversations, respectively. The latter has shorter utterances and more meaningless content, making the effects of our attention mechanisms vary across languages.

In particular, by applying generation with a prompt on ACE05-E$^+$, though the contents of dialogues are more focused on a topic, they also have some meaningless repetitive sentences which can not be seen as normal contextual texts. Applying self-attention to such contents would mess up token representations. For direct generation, the casualness and uncertainty of the generated contents make the influence of various attention mechanisms more consistent with our expectation.

On FOSAED-R, since user reviews are primarily informal texts, generated utterances may have jumbled characters and modal particles. And the nature of Weibo conversations make generated dialogues having some meaningless sentences. With direct generation, the sentence embeddings from [CLS] tokens may be useless, potentially making utterance attention impair performance. Generation with a prompt would yield consistent and coherent utterances, but the use of more attention mechanisms may confuse the model and make it more difficult to converge. When further training is conducted, generated dialogues are more domain-specific. However, as most of the utterances from the agent are less informative, it does not show a significant improvement in event detection.

Prompt	T	U	H
Prompt_1	72.3	72.6	73.0
Prompt_2	72.0	72.1	71.3
Prompt_3	71.6	73.5	71.8
Prompt_4	71.8	72.2	73.3

Table 5: Trig-C results(%) on ACE05-E$^+$ with different prompts to generate dialogues.

4.4 Effect of Different Prompts

We design four simple prompts to generate dialogues: (1) *What happened?* (2) *What happened in the previous sentence?* (3) *What event does the previous sentence describe?* (4) *Describe the event in the previous sentence.* The results on ACE05-E$^+$ are shown in Table 5.

Prompt_1 for generation and direct generation have the same trend under different attention mechanisms, as Prompt_1 is less topic-specific. However, it works better than direct generation. Prompt_2 and Prompt_3 work similarly, with Prompt_3 being slightly better than Prompt_2. Both of them add a phrase *in the previous sentence* to limit the scope of generated dialogues. Prompt_4 is the declarative form of Prompt_3, which imposes fewer constraints than the interrogative form.

4.5 Exploration of Generated Dialogues

To reveal the quality of generated dialogues and how the dialogue-based explanation impacts event detection, we heuristically design a feature ***p*(consistent)** to quantify the consistency of dialogues, which is defined as the percentage of generated dialogues consistent with the original sentences. This indicator intuitively specifies that if a sentence contains events, the generated dialogue

Generation	Indicator	ACE05-E$^+$	FOSAED-R
Direct	Length	54.6	62.1
	p(event)	11.9	19.5
	p(no-event)	93.2	72.2
	p(consistent)	58.0	30.7
Prompt_3	Length	60.9	79.2
	p(event)	21.2	24.0
	p(no-event)	80.4	71.1
	p(consistent)	54.7	34.0
Further	Length	-	134.6
	p(event)	-	41.1
	p(no-event)	-	26.7
	p(consistent)	-	38.1

Table 6: Heuristic exploration of different dialogue generation methods based on BERT and four indicators. The number of generated utterances is set to five.

Generation	Indicator	Context	Dialogue
Direct	Trig-C	70.6	70.9
	p(event)	22.5	11.9
	p(no-event)	50.4	93.2
	p(consistent)	38.3	58.0
Prompt_3	Trig-C	70.6	71.1
	p(event)	23.5	21.2
	p(no-event)	49.1	80.4
	p(consistent)	38.0	54.7

Table 7: Experiments of using plain narrative contexts or dialogues as additional information on ACE05-E$^+$. Five generated utterances are used for dialogue, and the number of generated tokens is set to the average token length of the five utterances for narrative contexts.

should contain all events in this sentence; if a sentence has no events, the generated dialogue would also has no events. The indicator can be divided into two sub-indicators ***p*(event)** and ***p*(no-event)**. p(event) indicates the number of generated dialogues containing all events in the original sentences as a percentage to the number of sentences with events. And p(no-event) indicates the number of generated dialogues having no events as a percentage to the number of sentences without events. We employ a BERT model to detect events in the generated dialogues consisting of five utterances. The average token length of generated dialogues is also used as a simple feature. It is noteworthy that these four indicators do not reflect true fluency of sentences and information intensity due to the inaccuracy of the BERT model, but they still provide a uniform quantitative metric for a relatively fair comparison.

Intuitively, a generation method producing more consistent dialogues should have higher scores on

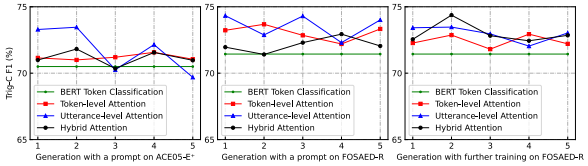

Figure 5: Effect of number of utterance on ACE05-E$^+$ with dialogue generation with a prompt and on FOSAED-R with dialogue generation with a prompt along with further training and generation.

p(consistent), p(event), and p(no-event). As reflected in Table 6, on ACE05-E$^+$, dialogues generated with Prompt_3 have higher p(event) and lower p(no-event) compared with dialogues generated directly. Since generation with a prompt can compensate for deficiencies in the structure and introduce prior knowledge from the prompt, it can generate more event-related dialogues, while more noise would be introduced. Combining the performance on event detection, we conclude that p(event) is a more crucial factor on the final results, however, a smaller p(no-event) may bring more noise, confusing the model and making it harder to converge. On FOSAED-R, generation with a prompt obtains better results on p(consistent) than generated directly, while generation after further training yields the best. Nevertheless, generation after further training significantly increases the average length of dialogues due to nonsensical utterances from an agent, which introduces unnecessary noise, thus giving limited improvement.

4.6 Comparison Between Dialogues and Narrative Contexts

To investigate how dialogue-based explanation differs from narrative contexts, we experiment by comparing the effects between using narrative contexts (generated by GPT-2) and dialogues on ACE05-E$^+$. Since the structure of contexts generated by GPT-2 is confusing and difficult to split for utilization, we only use token-level attention to derive results. For a dialogue, we use five generated utterances. For a narrative context, the number of generated tokens is set to the average token length of the five utterances.

As shown in Table 7, narrative contexts have slightly higher p(event) compared with dialogues, both generation directly and with a prompt, however, p(no-event) and p(consistent) are significantly lower. This observation suggests that narrative contexts are less consistent with original sentences.

Potential Issues	Generated Dialogues
Imbalanced Data in ACE05-E$^+$	"User: The Pakistani supreme court last year ***[aquitted | Justice:Acquit]*** Ayub Masih, who had denied uttering any word against Islam or the Prophet Mohammad. What happened?","Bot: He was sentenced to death.","Bot: He got life."
Multiple Events in ACE05-E$^+$	"User: Police **[*arrested* | *Justice:Arrest-Jail*]** her in Abilene, Texas, Saturday where she had ***[moved | Movement:Transport]*** with a friend June 2. What event does the previous sentence describe?", "Bot: I think it's a reference to the fact that she was arrested in Abillene, Texas, and then she was found in Abille, Texas.", "Bot: I thought it was a reference that she had been arrested in Texas, but was found at a different location."
Multiple Events in FOSAED-R	"用户:鱼都[没有煮熟|食品不熟]，一股[腥|感官异常]味，[恶心|身体不适]死了", "客服:有异味是吗亲亲,亲亲您方便把餐品照片发给商家吗", "用户:鱼鳞都没有刮干净,鱼腥味太重" ("User: The fish is **[*undercooked* | *Undercooked*]** and smells ***[fishy | Abnormalities]***, I feel ***[sick and disgusted | Uncomfortable]***","Agent: Is there a peculiar smell, my sweetie honey? Is it convenient for you to send the pictures of the food to the merchant","User: The fish scales are not even scraped clean, and the fishy smell is too strong")

Table 8: Case study for some potential issues in ACE05-E$^+$ and FOSAED-R, we compare DESED to BERT token classification based on original sentences as the baseline. The original sentence is the first utterance from User. The other utterances are generated. The format of the trigger and event is represented as **[*Trigger* | *Event_Type*]**, where color in red means that DESED can recognize but the baseline cannot, while color in black means that both the baseline and DESED can recognize.

The better performance using the generated dialogues also illustrates the superiority of dialogue-based explanation. Another advantage of dialogues over narrative contexts is that each utterance in a dialogue is an independent semantic unit that requires no additional segmentation, which is essential for various attention mechanisms.

4.7 Effect of Number of Utterance

Figure 5 shows the effect of number of utterance on ACE05-E$^+$ and FOSAED-R. On ACE05-E$^+$, we use Prompt_3. While on FOSAED-R, we use the prompt: 这句话描述了什么事件？ (which has the same English meaning as Prompt_3).

From the results, we can observe that: compared with the token-level attention, the utterance-level attention has a greater fluctuation on the number of utterance, and the hybrid attention is a fusion of them. Due to the randomness of dialogue generation, higher quality generated dialogues are more beneficial than dialogues with more utterances. After further training, the knowledge of the dialogue generation model is limited to a specific domain, thus having a smoother performance.

4.8 Case Study

We conduct a case study to further show the effectiveness of DESED intuitively in Table 8.

There exists data imbalance problem in ACE05-E$^+$ (e.g. Justice:Acquit only accounts for 1.1% of all events in the training set). Additional dialogue information can be utilized as an effective semantic complement for rare events. For a sentence with multiple events, while general methods may have difficulties capturing the association between events, DESED can further discover multiple events through generated dialogues.

5 Conclusion

In this paper, we propose a new paradigm, dialogue-based explanation, to enhance sentence semantics for sentence-level event detection. We propose three conceptually simple methods to generate dialogues for given original sentences, which concentrate on casual dialogues, focused dialogues and domain-specific dialogues respectively. To make effective use of generated dialogues, we design hybrid attention mechanisms at different levels of granularity. Extensive experiments and analyses show that DESED has promising performance on event detection. In the future, we are interested in generating dialogue-based explanation in a more controllable way and extending dialogue-based explanation to other tasks.

Acknowledgements

We would like to thank all reviewers for their insightful comments and valuable suggestions. This research is supported by the National Key Research and Development Program of China (No.2018YFB1403302).

References

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

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35.

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 (ACL-IJCNLP)*.

R. George Doddington, Alexis Mitchell, A. Mark Przybocki, A. Lance Ramshaw, Stephanie Strassel, and M. Ralph Weischedel. 2004. The automatic content extraction (ace) program - tasks, data, and evaluation. *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*.

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)*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. Degree: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Jacob Devlin Ming-Wei Chang Kenton and Lee 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 (NAACL-HLT)*.

Sawan Kumar and Partha Talukdar. 2021. Reordering examples helps during priming-based few-shot learning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP*.

D. John Lafferty, Andrew McCallum, and C. N. Fernando Pereira. 2002. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 19th International Conference on Machine Learning (ICML)*.

Dong-Ho Lee, Mahak Agarwal, Akshen Kadakia, Jay Pujara, and Xiang Ren. 2021. Good examples make a faster learner: Simple demonstration-based learning for low-resource ner. *arXiv preprint arXiv:2110.08454*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020. Event extraction as multi-turn question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (EMNLP)*.

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 (ACL)*.

Sha Li, Liyuan Liu, Yiqing Xie, Heng Ji, and Jiawei Han. 2022. Piled: An identify-and-localize framework for few-shot event detection. *arXiv preprint arXiv:2202.07615*.

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 (ACL)*.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xiao Liu, Zhunchen Luo, and He-Yan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information

extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.

David McClosky, Mihai Surdeanu, and Christopher D Manning. 2011. Event extraction as dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations (ICLR)*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

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

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Minh Van Nguyen, Viet Lai, and Thien Huu Nguyen. 2021. Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

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)*.

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

Yida Wang, Pei Ke, Yinhe Zheng, Kaili Huang, Yong Jiang, Xiaoyan Zhu, and Minlie Huang. 2020. A large-scale chinese short-text conversation dataset. In *CCF International Conference on Natural Language Processing and Chinese Computing (NLPCC)*.

Haoran Yan, Xiaolong Jin, Xiangbin Meng, Jiafeng Guo, and Xueqi Cheng. 2019. Event detection with multi-order graph convolution and aggregated attention. 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)*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL-SD)*.