Modeling Document-Level Context for Event Detection via Important Context Selection

Amir Pouran Ben Veyseh[1], **Minh Van Nguyen**[1],
Nghia Ngo Trung[2], **Bonan Min**[3], **and Thien Huu Nguyen**[1]
[1] Dept. of Computer and Information Science, University of Oregon, Eugene, OR, USA
[2] VinAI Research, Vietnam [3] Raytheon BBN Technologies, USA
{apouranb,minhnv,thien}@cs.uoregon.edu,
v.nghiant66@vinai.io,bonan.min@raytheon.com

Abstract

Event Detection (ED) aims to recognize and classify trigger words of events in text. The recent progress has featured advanced transformer-based language models (e.g., BERT) as a critical component in state-of-the-art models for ED. However, the length limit for input texts is a barrier for such ED models as they cannot encode long-range document-level context that has been shown to be beneficial for ED. To address this issue, we propose a novel method to model document-level context with BERT for ED that dynamically selects relevant sentences in the document for the event prediction of the target sentence. The target sentence will be then augmented with the selected sentences and consumed entirely by BERT for improved representation learning for ED. To this end, the REINFORCE algorithm is employed to train the relevant sentence selection for ED. Several information types are then introduced to form the reward function for the training process, including ED performance, sentence similarity, and discourse relations. Our extensive experiments on multiple benchmark datasets reveal the effectiveness of the proposed model, leading to new state-of-the-art performance.

1 Introduction

Event Detection (ED) is one of the fundamental tasks for Information Extraction. Its goal is to identify the word(s) in text that most clearly evoke events and classifying them into predefined event types (event trigger words). For instance, in the input sentence "*After the scandal, David James was fired from the company.*", an ED system needs to recognize the word "*fired*" as an event trigger and predict its event type as *End-Position*.

The early methods for ED have involved feature-based models (Ahn, 2006; Liao and Grishman, 2010a; Miwa et al., 2014) while recent work has featured deep learning methods (Nguyen and Grishman, 2015; Chen et al., 2015; Lin et al., 2020; Nguyen et al., 2021). Similar to other NLP tasks, the current best systems for ED leverage transformer-based language models, e.g., BERT (Devlin et al., 2019), as a critical encoding component to achieve state-of-the-art performance (Lai et al., 2020b; Lin et al., 2020). As such, most of the current transformer-based models for ED only focus on sentence-level context in which the scope of context to predict event type for each word is limited to the host sentence (Lu et al., 2019; Wang et al., 2019). However, it has been shown that document-level context also provides important information for deep learning models for ED (Chen et al., 2018; Zhao et al., 2018). For instance, in the document "*The troops were retreating cautiously. He was shocked after he heard "Fire!".*", to correctly predict *Attack* as the type of the event evoked by "*Fire*" (i.e., avoiding the confusion with the event type *End-Position*), it is necessary to consider the previous sentence with the important context words of "*troops*" and "*retreating*". Consequently, in this work, we aim to develop a transformer-based model that can effectively encode document context to boost the performance for ED.

There have been a few recent work that applies BERT-based models for document-level ED (Wang et al., 2020b; Trong et al., 2020). However, as BERT can only encode input texts with up to 512 sub-tokens (due to the quadratic self-attention complexity), current BERT-based document-level models for ED has only constrained their applications to document context that can fit into the BERT length limit. The typical approaches involve only considering short documents (Wang et al., 2020b) or truncating long documents (Trong et al., 2020), as also done for other tasks (Schweter and Akbik, 2020; Luoma and Pyysalo, 2020). These models are thus unable to encode long-range dependencies in document context that go beyond the length limit of BERT to further improve the ED performance.

To alleviate the length limit for BERT-based ED

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 5403–5413
November 7–11, 2021. ©2021 Association for Computational Linguistics

models, two major approaches from other NLP tasks can be considered: (1) Architecture Change for Self-Attention (Zaheer et al., 2020; Beltagy et al., 2020; Kitaev et al., 2020): In this group, the vanilla self-attention of transformer is replaced with some variant mechanism such as sparse self-attention (Zaheer et al., 2020) that can encode larger context with the same complexity as the original transformer (e.g., Longformer (Beltagy et al., 2020) can consume 4096 sub-tokens compared to 512 sub-tokens in BERT); and (2) Hierarchical Design (Adhikari et al., 2019; Jörke et al., 2020): These works employ the standard transformer-based language models to encode the context with certain length limit. For larger context, they combine the transformer model with other deep architectures to construct the final model for specific tasks. For instance, Adhikari et al. (2019) split the documents into multiple chunks (to fit the length limit of BERT) and send the BERT-based representations of the chunks into a recurrent neural network for document modeling. However, both solutions have their own drawbacks that might not be suitable for ED. For the former, the transformers with variants of vanilla self-attention are still limited to a certain length (e.g., 4096 for Longformer), thus unable to encode important contexts that are farther away in the documents for ED. Moreover, due to the changes in the self-attention mechanism, such variants generally lead to lower performance for NLP tasks compared to vanilla transformer (Beltagy et al., 2020). For the latter, standard transformer architectures are still employed to represent chunks of input texts to the extent of their length limits, thus failing to benefit from the ability of transformer to consume the entire input texts and capture long-range dependencies of documents for representation learning. In particular, the self-attention component of the models in this case is still bounded to the length limit and cannot directly capture the dependencies over the entire document.

To this end, to develop an effective BERT-based document-level model for ED, our motivation in this work is to explicitly select only important/relevant parts of a document for a given sentence and feed those context into BERT for improved representation learning. One the one hand, context selection allows the model to compress an input document into a smaller chunk with important context that can fit entirely into the length limit of BERT to better exploit the capacity of BERT for representation learning. Further, the ignorance of irrelevant context in the input for BERT might also reduce noise and improve representation vectors for ED. In particular, given an input sentence S_i in a document for event prediction, our method seeks to select a subset of sentences for S_i in the document that contain the most important context for the event prediction of S_i. The subset is also constrained so its concatenation with S_i (i.e., the compressed document) can fall within the length limit of BERT, thus enabling BERT to encode the compressed document entirely.

As such, the key question in our model is how to design the sentence selection to facilitate the encoding of important context in documents with BERT for ED. To this end, we argue that the important context for an input sentence S_i in a document should be determined by the improved performance of BERT for ED on S_i when S_i is augmented with the context. As such, we propose to use the policy-gradient method REINFORCE (Williams, 1992) to guide the context selection module using the performance of BERT for ED as the reward. In addition, we introduce auxiliary rewards based on linguistic intuition (i.e., semantic and discourse relations between the input sentence S_i and selected context sentences) to enhance the selection process. Our extensive experiments on benchmark datasets show that the proposed method can achieve state-of-the-art results for ED on both the sequence-labeling and the word-classification formulations.

2 Model

In the literature, ED has been formulated as sequence-labeling (Lin et al., 2020; Nguyen et al., 2021) or word-classification (Nguyen and Grishman, 2015) problems. In this work, we explore both formulations of the task. Formally, given the input sentence $S_i = [w_1, w_2, \ldots, w_n]$ (with n words) from the document $D = [S_1, S_2, \ldots, S_N]$ (with N sentences), the goal is to recognize and classify event triggers in S_i, leveraging the broader context in D. In the sequence-labeling formulation (Lin et al., 2020), as event triggers are allowed to involve multiple words, ED aims to assign a label y_t to each word $w_t \in S_i$ (using the BIO annotation schema) so the label sequence $Y = [y_1, y_2, \ldots, y_n]$ can capture event trigger boundaries and types in S_i. For the word-classification formulation (Chen et al., 2015; Nguyen and Grishman, 2015), the model additionally has an index t for the trigger candidate word $w_t \in S_i$ as an input and the goal is to predict the event type that w_t triggers (a multi-class classification problem). Here, we also include a special type *None* to indicate that a word is not an event trigger. As such, in the word-classification setting, the model performs separate event type predictions for each word in the input texts.

As discussed in the introduction, our goal is to design a document-level model for ED that can effectively leverage the representation learning ability of BERT (i.e., entirely encoding important context sentences for S_i in ED using the powerful BERT and self-attention). To this end, we propose to select a set C_i that contains the most important context sentences for the event prediction of S_i in D, i.e., $C_i \subset S_{context} = \{S_j \in D | j \neq i\}$. The target sentence S_i will be augmented with the sentences in C_i to form a shorter document D' (i.e., $D' = \{S_i\} \cup C_i$) for D. In our model, D' will be constrained to fall within the length limit of BERT, thus allowing BERT to consume D' entirely for improved representation learning for ED. In this way, the important context with arbitrary distance from S_i in D can be reached and packed into D' to allow effective document-context modeling with BERT for ED. In the rest of this section, for brevity, we remove the subscript i from C_i. Our model for ED in this work consists of two major components: (I) Prediction Model: This model consumes the shorter document D' and perform event type prediction for S_i, and (II) Context Selection: This component selects the important context sentence set C to perform ED for S_i.

2.1 Prediction Model

The prediction model M^{ED} consumes the set of important context sentences C to perform ED for the target sentence S_i. We will first describe the architecture for the prediction model in this section; the construction of C will be discussed later. As such, we first divide the set of selected sentences C into two subsets LC and RC for the sentences on the left and right context of S_i in D, i.e., $LC = \{S_j \in C | j < i\}$ and $RC = \{S_j \in C | j > i\}$. Next, the sentences in LC and RC are concatenated (following their orders in D) to form the two word sequences $[w_1^{LC}, w_2^{LC}, \ldots, w_{n_{LC}}^{LC}]$ and $[w_1^{RC}, w_2^{RC}, \ldots, w_{n_{RC}}^{RC}]$ where n_{LC} and n_{RC} are the number of words in LC and RC respectively. Afterward, we feed the context-augmented text for S_i, i.e., the short document $D' = [[CLS], w_1^{LC}, w_2^{LC}, \ldots, w_{n_{LC}}^{LC}, [SEP], w_1, w_2, \ldots, w_n, [SEP], w_1^{RC}, w_2^{RC}, \ldots, w_{n_{RC}}^{RC}]$, into BERT for representation computation. This is possible as the number of tokens in D' will be constrained to follow the length limit of BERT. We use the hidden states of the last layer of the BERT model to represent the input tokens: $E = [e_{CLS}, e_1^{LC}, e_2^{LC}, \ldots, e_{n_{LC}}^{LC}, e_{SEP}, e_1, e_2, \ldots, e_n, e_{SEP}, e_1^{RC}, e_2^{RC}, \ldots, e_{n_{RC}}^{RC}]$. Here, for tokens consisting of multiple word-pieces, we take the average of the hidden states of the word-pieces to obtain the representation vectors for the tokens. Finally, depending on the task formulation, we send the representations of the word(s) w_t (for word-classification) or $[w_1, w_2, \ldots, w_n]$ (for sequence-labeling) to a two-layer feed-forward layer FF followed by a the softmax function σ to obtain label probability distributions[1]: $P(\cdot | S_i, D', t) = \sigma(FF(e_t))$.

We utilize the negative log-likelihood as the training loss for M^{ED}. In particular, for word classification, the loss function is: $\mathcal{L}_{pred} = -\log(P(l^* | S_i, D', t))$ while those for sequence labeling is: $\mathcal{L}_{pred} = -\frac{1}{n}\sum_{j=1}^{n} \log(P(y_j^* | S_i, D', j))$. Here, l^* is the golden event type for $w_t \in S_i$ in the word-classification formulation while y_j^* is the golden BIO label for $w_j \in S_i$ in sequence labeling.

2.2 Context Selection

To select the most important context sentences C for S_i, our intuition is that a sentence S_j in D is important for the event prediction of S_i if including S_j into the short document D' for the base model M^{ED} can improve the performance for M^{ED} on S_i. In particular, to prepare the sentences for context selection, we first employ BERT$_{base}$ to obtain a representation vector for each sentence $S_j \in S_{context}$. As such, to customize the representation of S_j for S_i (i.e., our target sentence for ED), we concatenate the two sentences before feeding them to the BERT$_{base}$ model. Concretely, the input to the BERT$_{base}$ encoder is: $[CLS], w_1^j, w_2^j, \ldots, w_{n_j}^j, [SEP], w_1, w_2, \ldots, w_n$ where $S_j = [w_1^j, w_2^j, \ldots, w_{n_j}^j$ with n_j words. The representation of the $[CLS]$ token from the last layer of BERT$_{base}$ is then employed to represent S_j, i.e., denoted by x_j. Next, the representation

[1] For sequence-labeling, the feed-forward layer consumes the representation of every word $w_t \in S_i$ separately.

vectors $X = \{x_j | S_j \in S_{context}\}$ will be used by subsequent components to select important sentences for S_i. Note that the BERT$_{base}$ model for this context selection component is different from those used for the prediction model M^{ED} to facilitate the customization for each component.

Before describing the sentence selection process, we note that the number of words from the selected sentences in C cannot exceed $l_i = 512 - |S_i|$ as we want to make sure that the short document D' can fit into the BERT length limit. We design an iterative process that select important context sentences for S_i via multiple steps. At step $k+1$ in the process ($k \geq 0$), a sentence $S_{i_{k+1}}$ is chosen over the set of sentences that has not been selected in $S_{context}$, i.e., $S^k_{context} = S_{context} \setminus \{S_{i_1}, \ldots, S_{i_k}\}$, conditioning on the previously selected sentences $S_{i_1}, \ldots, S_{i_k}$. As such, we employ a Long Sort-Term Memory Network (LSTM) $LSTM$ to obtain representation vectors to summarize the sentences that have been chosen in prior steps. At step 0, the initial hidden state h_0 for $LSTM$ is set to zero. At step $k+1$, we use the hidden state h_k of $LSTM$ from prior step as a summarization for the sentences selected before. Afterward, we compute a selection score sc^{k+1}_j for each sentence $S_j \in S^k_{context}$ based on the representation vector x_j of S_j in X and h_k: $sc^{k+1}_j = sigmoid(G([x_j : h_k]))$ where G is a two-layer feed-forward network.

The sentence S_{j^*} with highest selection score, i.e., $S_{j^*} = argmax_{S_j \in S^k_{context}} sc^{k+1}_j$, is then considered for selection at this step. In particular, if selecting S_{j^*} causes the number of words in the selected sentences so far (i.e., $|S_{j^*}| + \sum_{q=1}^{k} |S_{i_q}|$) exceeds the limit l_i, the selection process stops and S_{j^*} is not included in the selected set C (i.e., $C = \{S_{i_1}, \ldots, S_{i_k}\}$ in this case). Otherwise, the selection process continues to the next step and S_{j^*} will be chosen and included in C (i.e., $S_{i_{k+1}} = S_{j^*}$). The hidden state of $LSTM$ is also updated for the current step, i.e., $h_{k+1} = LSTM(h_k, x_{j^*})$, to prepare for the continuation of sentence selection.

2.3 Selection Training

As motivated in the introduction, a context sentence $S_j \in S_{context}$ is considered as important for the event prediction of S_i if augmenting S_i with S_j can improve the ED performance for the model. As such, we employ the performance of the prediction model M^{ED} on S_i (i.e., obtained by ruining M^{ED} over the augmented short document D') as the training signal to guild the sentence selection process. In particular, we employ the REINFORCE algorithm (Williams, 1992) to facilitates the use of the ED performance of M^{ED} as the reward to train the context selection in our model. In addition, REINFORCE allows the incorporation of other information into the reward function to better supervise the training process. As such, given the selected context sentences in C, we consider the following information for the reward in our model:

(1) **Task-level Reward** R^{task}_i: This reward is based on the performance of the prediction model M^{ED} for ED on S_i. To measure the impact of the selected context C, M^{ED} is operated on the augmented short document D'. In particular, for the sequence-labeling setting, we use F1 score for the performance-based reward while accuracy is employed for those in the word-classification setting.

(2) **Semantics-level Reward** R^{sim}_i: In this reward, we propose to prefer context sentences that are semantically similar to the target sentence S_i for ED. The motivation is that similar/related context sentences (e.g., discussing the same events or topics) might provide more relevant information for the event prediction in S_i. To this end, we include the semantic similarity between S_i and the selected sentences in C as an auxiliary information for the reward to train our model. In particular, we first obtain representation vectors $\hat{S}_i$ and $\hat{C}$ for S_i and C by max-pooling the representation vectors for their corresponding words in E (i.e., produced by M^{ED} over D'): $\hat{S}_i = MAX\_POOL(e_1, e_2, \ldots, e_n)$ and $\hat{C} = MAX\_POOL(e^{LC}_1, \ldots, e^{LC}_{n_{LC}}, e^{RC}_1, \ldots, e^{RC}_{n_{RC}})$. Afterward, the dot-product (i.e., $\odot$) between $\hat{S}_i$ and $\hat{C}$ is used as the similarity-based reward for the model: $R^{sim}_i = \hat{S}_i \odot \hat{C}$.

(3) **Discourse-level Reward** R^{disc}_i: For this reward, we seek to promote context sentences that are most connected to S_i via the entity coreference relation in the selection process. Here, the motivation is that a context sentence S_j is more relevant/important for the event prediction on S_i if there are more entities mentioned in both S_j and S_j. In particular, as entities can serve as arguments in events, mentioning similar entities makes it more likely for S_j and S_i to refer to similar/related events, potentially leading to better relevance and usefulness of S_j for ED on S_i. To implement this idea, we first obtain all entity mentions and their coreference clusters in D using the

off-the-shelf tool Stanford CoreNLP. The discourse-level reward to capture our intuition is then computed by: $R_i^{disc} = \frac{1}{|C|}\sum_{S_j \in C} COR(S_j, S_i)$. In this formula, $COR(S_j, S_i)$ counts the number of entities that are mentioned in both S_j and S_i normalized by the total number of entities appearing in S_j and S_j (i.e., leveraging the detected entity mentions and clusters computed above).

Consequently, the overall reward function to train our context selection module with REINFORCE is $R_i(C) = \alpha R_i^{task} + \beta R_i^{sim} + \gamma R_i^{disc}$. For convenience, we treat C as the sequence of selected sentences from $S_{context}$, i.e., $C = S_{i_1} S_{i_2} \ldots S_{i_K}$ where K is the number of sentences in C. With REINFORCE, we seek to minimize the negative expected reward R_i over the possible choices of C: $\mathcal{L}_{sel} = -\mathbb{E}_{C' \sim P(C'|D,S_i)}[R_i(C')]$. The policy gradient is then estimated by: $\nabla \mathcal{L}_{sel} = -\mathbb{E}_{C' \sim P(C'|D,S_i)}[(R_i(C') - b)\nabla \log P(C'|D, S_i)]$. Using one roll-out sample, we further estimate $\nabla \mathcal{L}_{sel}$ via the selected sequence C: $\nabla \mathcal{L}_{sel} = -(R_i(C) - b)\nabla \log P(C|D, S_i)$ where b is the baseline to reduce variance. In this work, we obtain the baseline b via: $b = \frac{1}{|B|}\sum_{j=1}^{|B|} R_i(C^j)$, where $|B|$ is the mini-batch size and C^j is the selected sequence for the j-th sample in the mini-batch. Also, the probability of the selected sequence C is computed via: $P(C|D, S_i) = \prod_{k=0..K-1} P(S_{i_{k+1}}|D, S_i, S_{i_{\leq k}})$ where $S_{i_{\leq k}} = S_{i_1} \ldots S_{i_k}$ and $P(S_{i_{k+1}}|D, S_i, S_{i_{\leq k}})$ is obtained via the softmax function over selection scores for sentences in $S^k_{context}$ at selection step $k + 1$: $P(S_{i_{k+1}}|D, S_i, S_{i_{\leq k}}) = \exp(sc^{k+1}_{i_{k+1}}) / \sum_{S_j \in S^k_{context}} \exp(sc^{k+1}_j)$.

In this work, we train the prediction model M^{ED} and the context selection component in an alternate training fashion. Specifically, at each update step with one batch of data, we employ the current selection component to select important context sentences C for each target sentence S_i in the batch and form the short documents D'. The parameters for the prediction model M^{ED} will then be updated using the gradient of $\mathcal{L}_{pred}$ over the short documents for the current batch. This is followed by the update for the parameters of the selection component using the gradient of $\mathcal{L}_{sel}$ (i.e., performance of the current prediction model M^{ED} is used for the reward at this step). Finally, the same procedure applies for the test time where important context sentences will be first selected and then consumed by the prediction model to perform ED.

3 Experiments

Datasets & Baselines: To study the effectiveness of the proposed model, called Event Detection with Dynamic Document Context (ED3C), we evaluate its performance on two benchmark datasets **ACE 2005** (Walker et al., 2006) and **CySecED** (Trong et al., 2020). We choose these two datasets as they provide documents that are much longer than the input length limit for BERT$_{base}$, thus being more suitable to our focus on document-context modeling for ED[2]. We use full document context for the documents in these datasets.

ACE 2005 annotates 599 documents for 33 event types. We use the same data split and preprocessing as prior work (Lin et al., 2020; Lai et al., 2020b; Tong et al., 2020) for this dataset. The numbers of documents for the training/test/validation data are 529/40/30 respectively. In prior work, the ED problem on ACE 2005 has been addressed via both the sequence-labeling (Wang et al., 2020b; Lin et al., 2020) and word-classification (Lai et al., 2020b; Tong et al., 2020) formulations. The sequence-labeling formulation adheres to the original annotation in ACE 2005 to allow multiple words in event triggers. The word-classification formulation, in contrast, simplifies the problem by only concerning the single most important words in event triggers (Nguyen and Grishman, 2015). We use ACE 2005 to evaluate the systems on both formulations in this work. As such, we compare the performance of EC3C with the current state-of-the-art (SOTA) ED models on ACE 2005, i.e., **BiLSTM** (Wang et al., 2020b) (for sequence-labeling) and **EKD** (Tong et al., 2020) (for word-classification).

CySecED is a recent ED dataset that annotates 300 articles for 30 cybersecurity event types. We use the same data split proposed in the original paper (Trong et al., 2020) with 240/30/30 documents for training/test/validation data. Following prior work, we evaluate ED models on the word-classification setting. We compare ED3C with the current SOTA model on CySecED, i.e., **DEEB-RNN (word2vec)** (Trong et al., 2020; Zhao et al., 2018).

In addition to the aforementioned baselines, we also compare the proposed model with prior state-of-the-art document-level models for ED, includ-

[2] Most of documents in other ED datasets can be fit directly into the BERT$_{base}$ length limit. For instance, the proportions of documents with length larger than 512 are 9.1% in MAVEN (Wang et al., 2020b) and 0% in RAMS (Ebner et al., 2020).

Model	ACE 2005 (SL)			ACE 2005 (WP)			CySecED		
	P	R	F1	P	R	F1	P	R	F1
State-of-the-art	BiLSTM			EKD			DEEB-RNN (word2vec)		
	77.20	74.90	75.40	79.10	78.00	78.60	-	-	68.40
HBTNGMA	76.42	73.88	75.13	73.42	78.68	75.96	72.18	54.95	62.40
DEEB-RNN (BERT)	77.89	74.35	76.08	70.91	79.38	74.91	71.77	60.23	65.50
Hierarchical	77.19	73.02	75.05	72.84	79.46	76.01	71.42	56.32	62.98
Neighbor Sentences	77.92	71.62	74.64	70.31	80.98	75.27	70.13	55.54	61.99
BigBird	76.04	73.85	74.93	74.39	79.07	76.66	70.70	57.12	63.19
Reformer	76.91	73.60	75.22	79.32	73.24	76.16	72.91	54.43	62.33
Longformer	75.12	77.20	76.15	76.19	79.70	77.91	62.64	58.30	66.71
ED3C (ours)	80.31	76.04	**78.12**	75.13	83.51	**79.10**	75.14	65.57	**70.03**

Table 1: Performance of the models on ACE 2005 and CySecED test sets. SL and WP stands for sequence-labeling and word-classification. The proposed model is significantly better than baselines with $p < 0.01$.

ing **HBTNGMA** (Chen et al., 2018) that employs gated multi-level attention mechanism and **DEEB-RNN (BERT)** (Zhao et al., 2018) that uses a bidirectional RNN for encoding the sentences of the documents. Note that as BERT is not originally used in these models, we use their provided implementations and inject the BERT$_{base}$ model into the encoding components for a fairer comparison. Furthermore, we compare ED3C with other variants of pre-trained transformer-based language models capable of encoding input texts with longer length than BERT$_{base}$ limit. Specifically, we consider three commonly used language models: **BigBird** (Zaheer et al., 2020) which utilizes sparse self-attention, **Reformer** (Kitaev et al., 2020) which replaces dot-product attention with locality-sensitive hashing, and **Longformer** (Beltagy et al., 2020) that utilizes local self-attention together with global task-aware attention. Finally, we compare ED3C with two typical solutions for document encoding using BERT, i.e., **Hierarchical** (Adhikari et al., 2019) which splits a document into multiple chunks and encode them separately with BERT, a BiLSTM model is then employed to aggregate the representations of the chunks; and **Neighbor Sentences** (Schweter and Akbik, 2020) which augments the current sentence with sentences immediately appearing before and after the current sentence in the document for BERT-based encoding. Note that for BigBird, Reformer, Longformer, and "Neighbor Sentences", we use both left and right context sentences for augmentation. The corresponding quota for input length of a model (e.g., l_i for BERT) is divided evenly for the two types of context.

Hyperparameters: We tune the hyperparameters for the proposed model using a random search. All the hyperparameters are selected based on the F1 scores on the development set of the ACE 2005 dataset (with sequence-labeling setting). The same hyper-parameters from this fine-tuning are then applied for CySecED dataset and word-classification setting for consistency. In our model we use the BERT$_{base}$ to encode data; 300 dimensions for the hidden states of LSTM and 2 layers for feed-forward neural networks with 200 hidden dimensions. The trade-off parameters α, β and γ are set to 0.5, 0.1, 0.05, respectively. The learning rate is set to 3e-5 for the Adam optimizer and the batch size of 16 are employed during training.

Comparison: The performance of the models are shown in Table 1. There are several observations from this table. First, the modified transformer-based models Longformer outperforms the task-specific architectures that employ the standard BERT$_{base}$ model (e.g., HBTNGMA or Hierarchical). This is expected in that Longformer employs a variant of self-attention to entirely encode longer context than vanilla self-attention in the BERT (with limited input length), thus better capturing long-range dependencies in the context. The second observation is that the "Neighbor Sentences" baseline is generally worse than other models that attempt to exploit larger document context (e.g., the entire document in Hierachical and longer neighbor context in Longformer). This confirms the advantage of exploiting long document-level context for ED. Finally, the table shows that the proposed model is significantly better than all baselines and previous SOTA models on all settings and datasets. We attribute the superiority of ED3C to two impor-

Model	P	R	F1
ED3C (full)	81.23	76.60	78.85
- Sequential Selection	79.21	75.00	77.05
- Task-level Reward	78.12	74.10	76.06
- Semantics-level Reward	77.59	75.61	76.59
- Discourse-level Reward	78.91	75.40	77.12

Table 2: Performance of the ablated models on the development set of ACE 2005 (using the sequence-labeling formulation).

Model	P	R	F1
Only-Right	80.33	74.28	77.19
Only-Left	79.43	74.92	77.11
Neighbor Sentences	76.49	75.31	75.90
Most Similar	78.86	73.45	76.06
Highest Coreference	78.88	73.28	75.98
ED3C	81.23	76.60	78.85

Table 3: Model performance on the development set of ACE 2005 (with the sequence-labeling setting).

Document	Gold Label
According to the report, what distinguishes this matter from other flaws is the possibility for the hacker to *silently gain root access to these critical services*. Among all criticism, the researcher is pointing to the vendor as the main culprit. "It is understood that this is the responsibility of the vendors to protect user's sensitive data (including all their transactions for this matter) from the potentially **compromised** services," the researcher wrote in his report. We still closely monitor updates on this matter.	Discover. Backdoor
Although the report highlights that a simple *advertisement message within the system can expose the users to disclose their identities*, it discusses other vulnerabilities in detail. Some recent updates have potentially resolved the early concerns. However, the current weak **messaging** system is still the main reason for the unpopularity of the platform. Given the similar stories, it might take a few months even years for the developers to get the community's trust.	Discover. Social Engineering

Table 4: Case study in CySecED. Trigger words are shown in **red**. Their hosting sentences are the target sentences S_i for ED. Parts of the documents that are relevant to correctly infer the event types are shown in orange. The sentences containing those parts are successfully selected as important context sentences by ED3C.

tant characteristics that cannot be achieved in the baselines: (1) the ability to identify relevant context sentences at arbitrary length in the document to form shorter documents for BERT (thus also avoiding irrelevant sentences), (2) the capacity to encode important context entirely with BERT and self-attention to fully benefit from its representation learning for ED (i.e., avoiding the sacrifice of performance in exchange for efficiency due to the use of less powerful self-attention as in BigBird, Reformer, and Longformer).

Ablation Study: To better understand the proposed model, we conduct an ablation study in which we remove the major components of ED3C and report the performance of the remaining model on the development set of ACE 2005. We use the sequence-labeling setting in this study; however, the same patterns are observed for the word-classification setting. Specifically, we consider the following ablated models: (1) **- Sequential Selection (SS)**: In the selection component, we perform an sequence of selection steps where an LSTM network is employed to obtain a summarization of previously selected sentences, serving as a condition for selecting the next sentence for C. To assess the necessity of the multi-step selection with LSTM, we alter the selection process to a one-step selection. In particular, we only perform the sentence selection once where the the top T sentences with highest selection scores from the first step (i.e., sc_j^1 for $S_j \in S_{context}$) are selected to form the context C (i.e., eliminating LSTM). Here, K is determined such that the resulting short document D' can occupy the the input length limit of BERT$_{base}$ as such as possible; (2) **- Task-level Reward (TR)**: To study the impact of the performance reward, we remove R_i^{task} from the overall reward $R_i(C)$ for the context selection component; (3) **- Semantics-level Reward (SR)**: this model excludes the sentence similarity reward R_i^{sim} from the overall reward $R_i(C)$; (4) **- Discourse-level Reward (DR)**: Finally, this model eliminates the reward R_i^{disc} to prefer sentences with more discourse-level connections from $R_i(C)$.

The performance of the models are reported in Table 2. It is clear from the table that removing any of the components will hurt the model performance significantly, thus demonstrating their importance for the proposed method. In particular, among all ablated models, the removal of the task-level reward results in the most significant performance loss. This is expected as performance for ED is the most important and direct criteria to determine the salience of a context sentence in our task.

4 Analysis

Context Selection: To demonstrate the benefit of training the context sentence selection, we examine typical heuristics for choosing context sentences in a document D for the target sentence S_i in ED. As such, given the target sentence, these heuristics can directly suggest a set of context sentences (i.e., no training). The prediction model M^{ED} is then trained on the augmented texts of the suggested context and the target sentence for ED. Note that in these heuristics, context sentences are selected until the BERT$_{base}$ length limit is exceeded; the selected context sentences and target sentence are organized using their order in D to form the augmented short document D' for BERT. In particular, we explore the following selection heuristics in this section: (1) **Only-Right** and **Only-Left**: These approaches only augment the target sentence S_i with its right context sentences (i.e., $S_j \in D$ with $j > i$) and its left context sentences (i.e., $S_j \in D$ with $j < i$); (2) **Neighbor Sentences**: This approach is described in the baselines for ED3C above. It uses both left and right context sentences to augment S_i where the remaining space for context sentences in BERT length limit (i.e., $l_i = 512 - |S_i|$) is distributed evenly for the two context types; (3) **Most Similar**: In this baseline, we select the top context sentences S_j D that have the highest similarity with the target S_i. As such, we first obtain a representation vector for each sentence S_j by feeding S_j to the pre-trained BERT$_{base}$ model and using the hidden vector for the $[CLS]$ token in the last layer for this purpose. Afterward, the similarity between S_j and S_j is computed via the cosine similarity of their representation vectors; and (4) **Highest Coreference**: Finally, the selection strategy in this approach ranks context sentences $S_j \in S_{context}$ according to $COR(S_j, S_i)$, the normalized number of entities mentioned in both S_j and S_i (i.e., used to compute the reward R_i^{disc}. The context sentences with the highest $COR(S_j, S_i)$ scores are chosen for augmenting S_i.

The results of this analysis are provided in Table 3. As can be seen, the proposed model ED3C outperforms all heuristics-based baselines for context selection, thus demonstrating the benefit of a dynamic and learnable component for context selection in ED3C. Interestingly, selecting only right or left context achieves better results than “Neighbor Sentences”. It suggests that important context sentences might be far away from the target sentence in the document. Centering the target sentence in the context window thus might not be able to reach those sentences, further highlighting the advantage of ED3C on identifying relevant context that is arbitrarily far away for ED.

Case Study: To better understand the operation of ED3C, we manually analyze the examples in the development set of CySecED whose event types are not correctly predicted by the baselines (BigBird, Reformer, Longformer, and “Neighbor Sentence”), but can be successfully recognized by ED3C. Two examples for this category is presented in Table 4. The most important insight from our analysis is that the context sentence selection of ED3C allows the model to identify relevant sentences (i.e., as demonstrated in Table 4) and pack them into the short document D' for effective representation learning with BERT. This is in contrast to the baseline models (e.g., Longformer and “Neighbor Sentence”) that augment the target sentence S_i with all neighbor sentences. Noisy/irrelevant sentences might thus be included and impair induced representation vectors for ED. For instance, in Table 4, the immediately preceding sentences are not relevant to the event prediction of the target sentences, but are still introduced into the input texts for transformer-based models. Due to the trained context selection, ED3C can avoid those irrelevant sentences to perform ED correctly in these cases.

5 Related Work

ED has been approached with feature-based models earlier (Ahn, 2006; Patwardhan and Riloff, 2009; Liao and Grishman, 2010b; Hong et al., 2011; Li et al., 2013; Yang and Mitchell, 2016). Recently, deep learning (DL) methods are proved to be an effective approach for ED (Nguyen and Grishman, 2015; Chen et al., 2015; Nguyen et al., 2016a; Nguyen and Grishman, 2018; Sha et al., 2018; Zhang et al., 2019; Nguyen and Nguyen, 2019; Yang et al., 2019; Zhang et al., 2020; Le and Nguyen, 2021). Transformer-based language models such as BERT (Devlin et al., 2019) are the core components for current SOTA deep learning models for ED (Lai et al., 2020b; Veyseh et al., 2021). Recently, there have been growing interests to solve ED in the low-shot learning settings to improve the data efficiency of the models (Lai et al., 2020a, 2021). The majority of prior DL models for ED are restricted to sentence-level context. In recent years, encoding document context with DL has also been shown to be helpful for ED (Chen et al., 2018; Zhao et al., 2018; Zheng et al., 2019). However, existing DL methods for document-level ED have not employed transformer-based models, or only utilized them a limited manner (i.e., focusing on short documents or truncating long documents to fit the BERT length limit).

As described in the introduction, two main approaches have been explored to address the length limitation for transformer-based models: (1) transformers with modified self-attention (Beltagy et al., 2020; Kitaev et al., 2020; Wang et al., 2020a; Tay et al., 2020), and (2) truncating or hierarchical modeling (Adhikari et al., 2019; Luoma and Pyysalo, 2020; Jörke et al., 2020). The major drawback of these approaches involves the inability to capture important context of arbitrarily distance from the target and the failure to entirely consume important context with BERT and self-attention for effective representation learning. These issues are address by ED3C to boost the performance for ED.

6 Conclusion

We introduce a novel method, i.e., ED3C, to model document context for ED with BERT. Our model seeks to group important context sentences for the target sentence into a short document that can be entirely encoded by BERT to produce effective representation vectors for ED. As such, ED3C presents a context selection component that sequentially selects relevant sentences in the document based on LSTM. We use REINFORCE to train the context selection component in ED3C. Our extensive experiments demonstrate the effectiveness of ED3C with SOTA performance on benchmark datasets for ED. In future, we plan to apply the proposed model on other related tasks (e.g., Event Extraction).

Acknowledgments

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112 and the NSF grant CNS-1747798 to the IUCRC Center for Big Learning. This research is also based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the Better Extraction from Text Towards Enhanced Retrieval (BETTER) Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ARO, ODNI, IARPA, the Department of Defense, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Docbert: Bert for document classification. *arXiv preprint arXiv:1904.08398*.

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

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multipooling convolutional neural networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multilevel attention mechanisms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

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 Annual Meeting of the Association for Computational Linguistics (ACL)*.

Matthew Jörke, Jon Gillick, Matthew Sims, and David Bamman. 2020. Attending to long-distance document context for sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (EMNLP)*.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Viet Dac Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2020a. Exploiting the matching information in the support set for few shot event classification. In *Proceedings of the 24th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*.

Viet Dac Lai, Minh Van Nguyen, Thien Huu Nguyen, and Franck Dernoncourt. 2021. Graph learning regularization and transfer learning for few-shot event detection. In *Proceeddings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

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

Duong Le and Thien Huu Nguyen. 2021. Fine-grained event trigger detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Shasha Liao and Ralph Grishman. 2010a. Filtered ranking for bootstrapping in event extraction. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Shasha Liao and Ralph Grishman. 2010b. Using document level cross-event inference to improve event extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

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)*.

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 Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jouni Luoma and Sampo Pyysalo. 2020. Exploring cross-sentence contexts for named entity recognition with bert. *arXiv preprint arXiv:2006.01563*.

Makoto Miwa, Paul Thompson, Ioannis Korkontzelos, and Sophia Ananiadou. 2014. Comparable study of event extraction in newswire and biomedical domains. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Minh Van Nguyen, Viet Dac 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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

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

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Trung Minh Nguyen and Thien Huu Nguyen. 2019. One for all: Neural joint modeling of entities and events. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Siddharth Patwardhan and Ellen Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Stefan Schweter and Alan Akbik. 2020. Flert: Document-level features for named entity recognition. *arXiv preprint arXiv:2011.06993*.

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 *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. 2020. Synthesizer: Rethinking self-attention in transformer models. *arXiv preprint arXiv:2005.00743*.

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

Hieu Man Duc Trong, Duc Trong Le, Amir Pouran Ben Veyseh, Thuat Nguyen, and Thien Huu Nguyen. 2020. Introducing a new dataset for event detection in cybersecurity texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Amir Pouran Ben Veyseh, Viet Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2021. Unleash GPT-2 power for event detection. 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)*.

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

Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. 2020a. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

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

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

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Kluwer Academic*.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

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 Annual Meeting of the Association for Computational Linguistics (ACL)*.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*.

Junchi Zhang, Yanxia Qin, Yue Zhang, Mengchi Liu, and Donghong Ji. 2019. Extracting entities and events as a single task using a transition-based neural model. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Yunyan Zhang, Guangluan Xu, Yang Wang, Daoyu Lin, Feng Li, Chenglong Wu, Jingyuan Zhang, and Tinglei Huang. 2020. A question answering-based framework for one-step event argument extraction. In *IEEE Access, vol 8, 65420-65431*.

Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Document embedding enhanced event detection with hierarchical and supervised attention. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction. 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)*.