MLBiNet: A Cross-Sentence Collective Event Detection Network

Dongfang Lou[1,2] *, **Zhilin Liao** [1,2] *, **Shumin Deng**[1,2], **Ningyu Zhang**[1,2] †, **Huajun Chen**[1,2] †
[1] Zhejiang University & AZFT Joint Lab for Knowledge Engine
[2] Hangzhou Innovation Center, Zhejiang University
loudongfang2015@163.com, zhilinliao@yeah.net
{231sm,zhangningyu,huajunsir}@zju.edu.cn

Abstract

We consider the problem of collectively detecting multiple events, particularly in cross-sentence settings. The key to dealing with the problem is to encode semantic information and model event inter-dependency at a document-level. In this paper, we reformulate it as a Seq2Seq task and propose a **M**ulti-**L**ayer **Bi**directional **Net**work (MLBiNet) to capture the document-level association of events and semantic information simultaneously. Specifically, a bidirectional decoder is firstly devised to model event inter-dependency within a sentence when decoding the event tag vector sequence. Secondly, an information aggregation module is employed to aggregate sentence-level semantic and event tag information. Finally, we stack multiple bidirectional decoders and feed cross-sentence information, forming a multi-layer bidirectional tagging architecture to iteratively propagate information across sentences. We show that our approach provides significant improvement in performance compared to the current state-of-the-art results[1].

1 Introduction

Event detection (ED) is a crucial sub-task of event extraction, which aims to identify and classify event triggers. For instance, the document shown in Table 1, which contains six sentences $\{s_1, \ldots, s_6\}$, the ED system is required to identify four events: an *Injure* event triggered by "injuries", two *Attack* events triggered by "firing" and "fight", and a *Die* event triggered by "death".

Detecting event triggers from natural language text is a challenge task because of the following problems: a). **Sentence-level contextual representation and document-level information aggregation** (Chen et al., 2018; Zhao et al., 2018; Shen et al., 2020). In ACE 2005 corpus, the arguments of a single event instance may be scattered in multiple sentences (Zheng et al., 2019; Ebner et al., 2019), which indicates that document-level information aggregation is critical for ED task. What's more, a word in different contexts would express different meanings and trigger different events. For example, in Table 1, "firing" in s_3 means the action of firing guns (*Attack* event) or forcing somebody to leave their job (*End_Position* event). To specify its event type, cross-sentence information should be considered. b). **Intra-sentence and inter-sentence event inter-dependency modeling** (Liao and Grishman, 2010; Chen et al., 2018; Liu et al., 2018). For s_4 in Table 1, an *Attack* event is triggered by "fight", and a *Die* event is triggered by "death". This kind of event co-occurrence is common in ACE 2005 corpus, we investigated the dataset and found that about 44.4% of the triggers appeared in this way. The cross-sentence event co-occurrence shown in s_4 and s_3 is also very common. Therefore, modeling the sentence-level and document-level event inter-dependency is crucial for jointly detecting multiple events.

s_1: what a brave young woman
s_2: did you hear about the **injuries**[*Injure*] she sustained
s_3: did you hear about the **firing**[*Attack*] she did
s_4: she was going to **fight**[*Attack*] to the **death**[*Die*]
s_5: she was captured but she was one tough cookie
s_6: god bless here

Table 1: An example document in ACE 2005 corpus with cross-sentence semantic enhancement and event inter-dependency. Specifically, semantic information of s_2 provides latent information to enhance s_3, and *Attack* event in s_4 also contributes to s_3.

To address those issues, previous approaches (Chen et al., 2015; Nguyen et al., 2016; Liu et al., 2018; Yan et al., 2019; Liu et al., 2019; Zhang et al., 2019) mainly focused on sentence-level event de-

* Equal contribution and shared co-first authorship.
† Corresponding author.
[1]The code is available in https://github.com/zjunlp/DocED.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 4829–4839
August 1–6, 2021.

tection, neglecting the document-level event inter-dependency and semantic information. Some studies (Chen et al., 2018; Zhao et al., 2018) tried to integrate semantic information across sentences via the attention mechanism. For the document-level event inter-dependency modeling, Liao and Grishman (2010) extended the features with event types to capture dependencies between different events in a document. Although great progress has been made in ED task due to recent advances in deep learning, there is still no unified framework to model the document-level semantic information and event inter-dependency.

We try to analyze the ACE 2005 data to re-understand the challenges encountered in ED task. Firstly, we find that event detection is essentially a special Seq2Seq task, in which the source sequence is a given document or sentence, and the event tag sequence is target of task. Seq2Seq tasks can be effectively modeled via the RNN-based encoder-decoder framework, in which the encoder captures rich semantic information, while the decoder generates a sequence of target symbols with inter-dependency been captured. This separate encoder and decoder framework can correspondingly deal with the semantic aggregation and event inter-dependency modeling challenges in ED task. Secondly, for the propagation of cross-sentence information, we find that the relevant information is mainly stored in several neighboring sentences, while little is stored in distant sentences. For example, as shown in Table 1, it seems that s_2 and s_4 contribute more to s_3 than s_1 and s_5.

In this paper, we propose a novel **M**ulti-**L**ayer **Bi**directional **Net**work (MLBiNet) for ED task. A bidirectional decoder layer is firstly devised to decode the event tag vector corresponding to each token with forward and backward event inter-dependency been captured. Then, the event-related information in the sentence is summarized through a sentence information aggregation module. Finally, the multiple bidirectional tagging layers stacking mechanism is proposed to propagate cross-sentence information between adjacent sentences, and capture long-range information as the increasing of layers. We conducted experimental studies on ACE 2005 corpus to demonstrate its benefits in cross-sentence joint event detection. Our contributions are summarized as follows:

- We propose a novel bidirectional decoder model to explicitly capture bidirectional event inter-dependency within a sentence, alleviating long-range forgetting problem of traditional tagging structure;
- We propose a model called MLBiNet to propagate semantic and event inter-dependency information across sentences and detect multiple events collectively;
- We achieve the best performance (F_1 value) on ACE 2005 corpus, surpassing the state-of-the-art by 1.9 points.

2 Approach

Generally, event detection on ACE 2005 corpus is treated as a classification problem, which is to determine whether it forms a part of an event trigger. Specifically, for a given document $d = \{s_1, \ldots, s_n\}$, where $s_i = \{w_{i,1}, \ldots, w_{i,n_i}\}$ denotes the i-th sentence containing n_i tokens. We are required to predict the triggered event type sequence $y_i = \{y_{i,1}, \ldots, y_{i,n_i}\}$ based on contextual information of d. Without ambiguity, we omit the subscript i.

For a given sentence, the event tags corresponding to tokens are associated, which is important for collectively detecting multiple events (Chen et al., 2018; Liu et al., 2018). The way tokens are classified independently will miss the association. In order to capture the event inter-dependency, the sequential information of event tag should be retained. Intuitively, the ED task can be regarded as event tag sequence generation problem, which is essentially a Seq2Seq task. Specifically, the source sequence is a given document or sentence, and the event tag sequence to be generated is the target sequence. For instance, for sentence "did you hear about the injuries she sustained", the decoder model is required to generate a tag sequence $[O, O, O, O, O, B\_Injure, O, O]$, where "O" denotes that the corresponding token is not part of event trigger and "*B_Injure*" indicates an *Injure* event is triggered.

We introduce the RNN-based encoder-decoder framework for ED task, considering that it is an efficient solution for Seq2Seq tasks. And we propose a multi-layer bidirectional network called MLBiNet shown in Figure 1 to deal with the challenges in detecting multiple events collectively. The model framework consists of four components: the semantic encoder, the bidirectional decoder, the information aggregation module and stacking of mul-

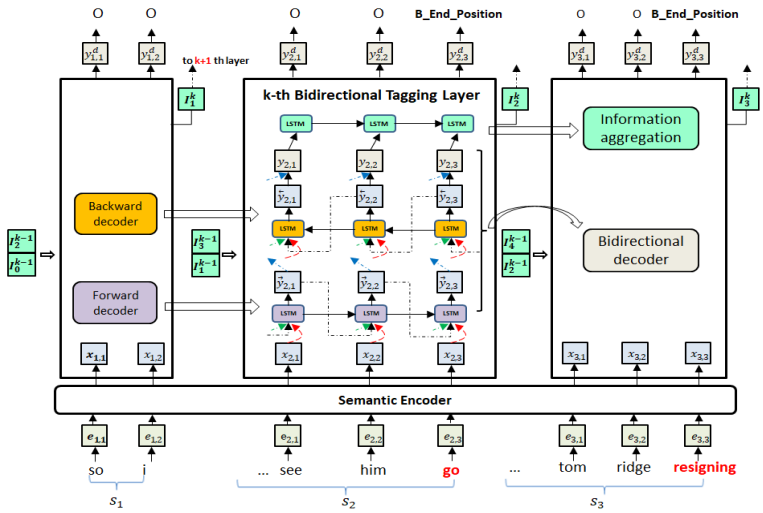

Figure 1: The architecture of our multi-layer bidirectional network (MLBiNet). The red arrow represents the input of semantic representation $\mathbf{x}_t$, the green arrow represents the input of adjacent sentences information $[\mathbf{I}_{i-1}^{k-1}; \mathbf{I}_{i+1}^{k-1}]$ integrated in the previous layer, and the blue arrow represents the input of forward event tag vector.

tiple bidirectional tagging layers. We firstly introduce the encoder-decoder framework and discuss its compatibility with the ED task.

2.1 Encoder–Decoder

The RNN-based encoder-decoder framework (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015; Gu et al., 2016) consists of two components: a) an encoder which converts the source sentence into a fixed length vector $\mathbf{c}$ and b) a decoder is to unfold the context vector $\mathbf{c}$ into the target sentence. As is formalized in (Gu et al., 2016), the source sentence s_i is converted into a fixed length vector $\mathbf{c}$ by the encoder RNN,

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, w_t), \ \mathbf{c} = \phi(\{\mathbf{h}_1, \ldots, \mathbf{h}_{n_i}\})$$

where f is the RNN function, $\{\mathbf{h}_t\}$ are the RNN states, w_t is the t-th token of source sentence, $\mathbf{c}$ is the so-called context vector, and ϕ summarizes the hidden states, e.g. choosing the last state $\mathbf{h}_{n_i}$. And the decoder RNN translates $\mathbf{c}$ into the target sentence according to:

$$\begin{aligned} \mathbf{s}_t &= f(y_{t-1}, \mathbf{s}_{t-1}, \mathbf{c}) \\ p(y_t|y_{<t}, s_i) &= g(y_{t-1}, \mathbf{s}_t, \mathbf{c}) \end{aligned} \quad (1)$$

where $\mathbf{s}_t$ is the state at time t, y_t is the predicted symbol at time t, g is a classifier over the vocabulary, and $y_{<t}$ denotes the history $\{y_1, \ldots, y_{t-1}\}$.

Studies (Bahdanau et al., 2015; Luong et al., 2015) have shown that summarizing the entire source sentence into a fixed length vector will limit the performance of the decoder. They introduced the attention mechanism to dynamically changing context vector $\mathbf{c}_t$ in the decoding process, where $\mathbf{c}_t$ can be uniformly expressed as

$$\mathbf{c}_t = \sum_{\tau=1}^{n_i} \alpha_{t\tau} \mathbf{h}_\tau \quad (2)$$

where $\alpha_{t\tau}$ is the contribution weight of τ-th source token's state to context vector at time t, $\mathbf{h}_\tau$ denotes the representation of τ-th token.

We introduce the encoder-decoder framework to model ED task, mainly considering the following advantages: a) the separate encoder module is flexible in fusing sentence-level and document-level semantic information and b) the RNN decoder model (1) can capture sequential event tag dependency as the predicted tag vectors before t will be used as input for predicting t-th symbol.

The encoder-decoder framework for ED task is slightly different from the general Seq2Seq task as follows: a) For ED task, the length of event tag sequence (target sequence) is known because its elements correspond one-to-one with tokens in the source sequence. However, the length of target sequence in the general Seq2Seq task is unknown. b) The vocabulary of decoder for ED task is a collection of event types, instead of words.

2.2 Semantic Encoder

In this module, we encode the sentence-level contextual information for each token with Bidirectional LSTM (BiLSTM) and self-attention mechanism. Firstly, each token is transformed into

comprehensive representation by concatenating its word embedding and NER type embedding. The word embedding matrix is pretrained by Skip-gram model (Mikolov et al., 2013), and the NER type embedding matrix is randomly initialized and updated in the training process. For a given token w_t, its embedded vector is denoted as $\mathbf{e}_t$.

We apply the BiLSTM (Zaremba and Sutskever, 2014) model for sentence-level semantic encoding, which can effectively capture sequential and contextual information for each token. The BiLSTM architecture is composed of a forward LSTM and a backward LSTM, i.e., $\overrightarrow{\mathbf{h}}_t = \overrightarrow{\text{LSTM}}(\overrightarrow{\mathbf{h}}_{t-1}, \mathbf{e}_t)$, $\overleftarrow{\mathbf{h}}_t = \overleftarrow{\text{LSTM}}(\overleftarrow{\mathbf{h}}_{t+1}, \mathbf{e}_t)$. After encoding, the contextual representation of each token is $\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t]$.

Attention mechanism between tokens within a sentence has been proven to further integrate long-range contextual semantic information. For each token w_t, its contextual representation is the weighted average of the semantic information of all tokens in the sentence. We apply the attention mechanism proposed by (Luong et al., 2015) with the weights derived by

$$\begin{aligned} \alpha_{t,j} &= \frac{\exp(z_{t,j})}{\sum_{m=1}^{n_i} \exp(z_{t,m})} \\ z_{t,m} &= \tanh(\mathbf{h}_t^\top W_{sa} \mathbf{h}_m + b_{sa}) \end{aligned} \tag{3}$$

And the contextual representation of w_t is $\mathbf{h}_t^a = \sum_{j=1}^{n_i} \alpha_{t,j} \mathbf{h}_j$. By concatenating its lexical embedding and contextual representation, we get the final comprehensive semantic representation of w_t as $\mathbf{x}_t = [\mathbf{h}_t^a; \mathbf{e}_t]$.

2.3 Bidirectional Decoder

The decoder layer for ED task is to generate a sequence of event tags corresponding to tokens. As is noted, the tag sequence (target sequence) elements and tokens (source sequence) are in one-to-one correspondence. Therefore, the context vector $\mathbf{c}$ shown in (1) and (2) can be personalized directly by $\mathbf{c}_t = \mathbf{x}_t$, which is equivalent to attention with degenerate weights. That is, $\alpha_{tt} = 1$ and $\alpha_{t\tau} = 0$, $\forall \tau \neq t$.

In traditional Seq2Seq tasks, the target sequence length is unknown during the inference process, so only the forward decoder is feasible. However, for the ED task, the length of the target sequence is known when given source sequence. Thus, we devise a bidirectional decoder to model event inter-dependency within a sentence.

Forward Decoder In addition to the semantic context vector $\mathbf{c}_t = \mathbf{x}_t$, the event information previously involved can help determine the event type triggered by t-th token. This kind of association can be captured by the forward decoder model:

$$\begin{aligned} \overrightarrow{\mathbf{s}}_t &= f_{\text{fw}}(\overrightarrow{\mathbf{y}}_{t-1}, \overrightarrow{\mathbf{s}}_{t-1}, \mathbf{x}_t) \\ \overrightarrow{\mathbf{y}}_t &= \tilde{f}(W_y \overrightarrow{\mathbf{s}}_t + b_y) \end{aligned} \tag{4}$$

where f_{fw} is the forward RNN, $\{\overrightarrow{\mathbf{s}}_t\}$ are the states of forward RNN, $\{\overrightarrow{\mathbf{y}}_t\}$ are the forward event tag vectors. Compared with general decoder (1), the classifier $g(\cdot)$ over vocabulary is replaced with a transformation $\tilde{f}(\cdot)$ (identity function, $\tanh$, sigmoid, etc.) to obtain the event tag vector.

Backward Decoder Considering the associated events may also be mentioned later, we devise a backward decoder to capture this kind of dependency as follows:

$$\begin{aligned} \overleftarrow{\mathbf{s}}_t &= f_{\text{bw}}(\overleftarrow{\mathbf{y}}_{t+1}, \overleftarrow{\mathbf{s}}_{t+1}, \mathbf{x}_t) \\ \overleftarrow{\mathbf{y}}_t &= \tilde{f}(W_y \overleftarrow{\mathbf{s}}_t + b_y) \end{aligned} \tag{5}$$

where f_{bw} is the backward RNN, $\{\overleftarrow{\mathbf{s}}_t\}$ are the states of backward RNN, $\{\overleftarrow{\mathbf{y}}_t\}$ are the backward event tag vectors.

Bidirectional Decoder By concatenating $\overrightarrow{\mathbf{y}}_t$ and $\overleftarrow{\mathbf{y}}_t$, we get the event tag vector $\mathbf{y}_t = [\overrightarrow{\mathbf{y}}_t; \overleftarrow{\mathbf{y}}_t]$ with bidirectional event inter-dependency been captured. The semantic and event-related entity information is also carried by $\mathbf{y}_t$ as $\mathbf{x}_t$ is an indirect input.

An alternative method modeling the sentence-level event inter-dependency called hierarchical tagging layer is proposed by (Chen et al., 2018). The bidirectional decoder is quite different from the hierarchical tagging layer as follows:

- The bidirectional decoder models event inter-dependency immediately by combining a forward and a backward decoder. The hierarchical tagging layer utilizes two forward decoders and the tag attention mechanism to capture bidirectional event inter-dependency.

- In the bidirectional decoder, the ED task is formalized as a special Seq2Seq task, which can simplify the event inter-dependency modeling problem and cross-sentence information propagation problem discussed below.

The bidirectional RNN decoder unfolds the event tag vector corresponding to each token, and captures the bidirectional event inter-dependency within the sentence. To propagate information across sentences, we need to firstly aggregate useful information of each sentence.

2.4 Information Aggregation

For current sentence s_i, the information we are concerned about can be summarized as recording which entities and tokens trigger which events. Thus, to summarize the information, we devise another LSTM layer (information aggregation module shown in Figure 1) with the event tag vector $\mathbf{y}_t$ as input. The information at t-th token is computed by

$$\tilde{\mathbf{I}}_t = \overrightarrow{\text{LSTM}}(\tilde{\mathbf{I}}_{t-1}, \mathbf{y}_t) \tag{6}$$

We choose the last state $\tilde{\mathbf{I}}_{n_i}$ as the summary information, which is $\mathbf{I}_i = \tilde{\mathbf{I}}_{n_i}$.

The sentence-level information aggregation module bridges the information across sentences, as the well-formalized information can be easily integrated into the decoding process of other sentences, enhancing the event-related signal.

2.5 Multi-Layer Bidirectional Network

In this module, we introduce a multiple bidirectional tagging layers stacking mechanism to aggregate information of adjacent sentences into the bidirectional decoder, and propagate information across sentences. The information $(\{\mathbf{y}_t\}, \mathbf{I}_i)$ obtained by the bidirectional decoder layer and information aggregation module has captured the event relevant information within a sentence. However, the cross-sentence information has not yet interacted. For a given sentence, as we can see in Table 1, its relevant information is mainly stored in several neighboring sentences, while distant sentences are rarely relevant. Thus, we propose to transmit the summarized sentence information $\mathbf{I}_i$ among adjacent sentences.

For the decoder framework shown in (4) and (5), the cross-sentence information can be integrated by extending the input with $\mathbf{I}_{i-1}$ and $\mathbf{I}_{i+1}$. Further, we introduce a multiple bidirectional tagging layers stacking mechanism shown in Figure 1 to iteratively aggregate information of adjacent sentences. The overall framework is named **M**ulti-**L**ayer **Bi**directional **Net**work (MLBiNet). As shown in Figure 1, a bidirectional tagging layer is composed of a bidirectional decoder and an information aggregation module. For sentence s_i, the outputs of k-th layer can be computed by

$$\begin{aligned}
\overrightarrow{\mathbf{s}}_t &= f_{\text{fw}}(\overrightarrow{\mathbf{y}}^k_{t-1}, \overrightarrow{\mathbf{s}}_{t-1}, \mathbf{x}_t, \mathbf{I}^{k-1}_{i-1}, \mathbf{I}^{k-1}_{i+1}) \\
\overleftarrow{\mathbf{s}}_t &= f_{\text{bw}}(\overleftarrow{\mathbf{y}}^k_{t+1}, \overleftarrow{\mathbf{s}}_{t+1}, \mathbf{x}_t, \mathbf{I}^{k-1}_{i-1}, \mathbf{I}^{k-1}_{i+1}) \\
\overrightarrow{\mathbf{y}}^k_t &= \tilde{f}(W_y \overrightarrow{\mathbf{s}}_t + b_y) \\
\overleftarrow{\mathbf{y}}^k_t &= \tilde{f}(W_y \overleftarrow{\mathbf{s}}_t + b_y) \\
\mathbf{y}^k_t &= [\overrightarrow{\mathbf{y}}^k_t; \overleftarrow{\mathbf{y}}^k_t]
\end{aligned} \tag{7}$$

where $\mathbf{I}^{k-1}_{i-1}$ is the sentence information of s_{i-1} aggregated in (k-1)-th layer, and $\{\mathbf{y}^k_t\}$ are event tag vectors obtained in k-th layer. The equation suggests that for each token of source sentence s_i, the input of cross-sentence information is identical $[\mathbf{I}^{k-1}_{i-1}, \mathbf{I}^{k-1}_{i+1}]$. It is reasonable as their cross-sentence information available is the same for each token of current sentence.

The iteration process shown in equation (7) is actually an evolutionary diffusion of the cross-sentence semantic and event information in the document. Specifically, in the first tagging layer, information of current sentence is effectively modeled by the bidirectional decoder and information aggregation module. In the second layer, information of adjacent sentences is propagated to current sentence by plugging in $\mathbf{I}^1_{i-1}$ and $\mathbf{I}^1_{i+1}$ to the decoder. In general, in the k-th ($k \geq 3$) layer, since s_{i-1} has captured the information of sentence s_{i-k+1} in the (k-1)-th layer, then s_i can obtain information in s_{i-k+1} by acquiring the information in s_{i-1}. Thus, as the number of decoder layers increases, the model will capture information from distant sentences. For K-layer bidirectional tagging model, the sentence information with the longest distance of K-1 can be captured.

We define the final event tag vector of w_t as the weighted sum of $\{\mathbf{y}^k_t\}_k$ in different layers, i.e., $\mathbf{y}^d_t = \sum_{k=1}^{K} \alpha^{k-1} \mathbf{y}^k_t$, where $\alpha \in (0, 1]$ is a weight decay parameter. It means that cross-sentence information can supplement to the current sentence, and the contribution gradually decreases as the distance increases when $\alpha < 1$.

We note that the parameters of bidirectional decoder and information aggregation module at different layers can be shared, because they encode and propagate the same structured information. In this paper, we set the parameters of different layers to be the same.

2.6 Loss Function

In order to train the networks, we minimize the negative log-likelihood loss function $J(\theta)$,

$$J(\theta) = -\sum_{d \in D} \sum_{s \in d} \sum_{w_t \in s} \log p(O_t^{y_t} | d; \theta) \quad (8)$$

where D denotes training documents set. The tag probability for token w_t is computed by

$$\begin{aligned} O_t &= W_o \mathbf{y}_t^d + b_o \\ p(O_t^j | d; \theta) &= \exp(O_t^j) / \sum_{m=1}^{M} \exp(O_t^m) \end{aligned} \quad (9)$$

where M is the number of event classes, $p(O_t^j | d; \theta)$ is the probability that assigning event type j to token w_t in document d when parameter is θ.

3 Experiments

3.1 Dataset and Settings

We performed extensive experimental studies on the ACE 2005 corpus to demonstrate the effectiveness of our method on ED task. It defines 33 types of events and an extra *NONE* type for the non-trigger tokens. We formalize it as a task to generate a sequence of 67-class event tag (with BIO tagging schema). The data splitting for training, validation and testing follows (Ji and Grishman, 2008; Chen et al., 2015; Liu et al., 2018; Chen et al., 2018; Huang and Ji, 2020), where the training set contains 529 documents, the validation set contains 30 documents and the remaining 40 documents are used as testing set.

We evaluated the performance of three multi-layer settings with 1-, 2- and 3-layer MLBiNet, respectively. We use the Adam (Kingma and Ba, 2017) for optimization. In all three settings, we cut every 8 consecutive sentences into a new document and padding when needed. Each sentence is truncated or padded to make it 50 in length. We set the dimension of word embedding as 100, the dimension of golden NER type and subtype embedding as 20. We set the dropout rate as 0.5 and penalty coefficient as $2 * 10^{-5}$ to avoid overfitting. The hidden size of semantic encoder layer and decoder layer is set to 100 and 200, respectively. The size of forward and backward event tag vectors is set to 100. And we set the batch size as 64, the learning rate as $5 * 10^{-4}$ with decay rate 0.99, the weight decay parameter α as 1.0. The results we report are the average of 10 trials.

Methods	P	R	F_1
DMCNN	75.6	63.6	69.1
HBTNGMA	77.9	69.1	73.3
JMEE	76.3	71.3	73.7
DMBERT-Boot	77.9	72.5	75.1
MOGANED	**79.5**	72.3	75.7
SS-VQ-VAE	75.7	77.8	76.7
MLBiNet (1-layer)	74.1	78.5	76.2
MLBiNet (2-layer)	74.2	**83.7**	**78.6**
MLBiNet (3-layer)	74.7	83.0	**78.6**

Table 2: Performance comparison of different methods on the test set with gold-standard entities.

3.2 Baselines

For comparison, we investigated the performance of the following state-of-the-art methods: 1) **DMCNN** (Chen et al., 2015), which extracts multiple events from one sentence with dynamic multi-pooling CNN; 2) **HBTNGMA** (Chen et al., 2018), which models sentence event inter-dependency via a hierarchical tagging model; 3) **JMEE** (Liu et al., 2018), which models the sentence-level event inter-dependency via a graph model of the sentence syntactic parsing graph; 4) **DMBERT-Boot** (Wang et al., 2019), which augments the training data with external unlabeled data by adversarial mechanism; 5) **MOGANED** (Yan et al., 2019), which uses graph convolution network with aggregative attention to explicitly model and aggregate multi-order syntactic representations; 6) **SS-VQ-VAE** (Huang and Ji, 2020), which learns to induct new event type by a semi-supervised vector quantized variational autoencoder framework, and fine-tunes with the pre-trained BERT-large model.

3.3 Overall Performance

Table 2 presents the overall performance comparison between different methods with gold-standard entities. As shown, under 2-layer and 3-layer settings, our proposed model MLBiNet achieves better performance, surpassing the current state-of-the-art by 1.9 points. More specifically, our models achieve higher recalls by at least 0.7, 5.9 and 5.2 points, respectively.

The powerful encoder of BERT pre-trained model (Devlin et al., 2018) has been proven to improve the performance of downstream NLP tasks. The 2-layer MLBiNet outperforms BERT-Boot (BERT-base) and SS-VQ-VAE (BERT-large) by 3.5 and 1.9 points, respectively. It proves the im-

Methods	1/1	1/n	all
DMCNN	74.3	50.9	69.1
HBTNGMA	78.4	59.5	73.3
JMEE	75.2	72.7	73.7
MLBiNet (1-layer)	77.9	75.1	76.2
MLBiNet (2-layer)	**80.6**	77.1	**78.6**
MLBiNet (3-layer)	80.3	**77.4**	**78.6**

Table 3: System Performance on Single Event Sentences (1/1) and Multiple Event Sentences (1/n). 1/1 means one sentence that has one event; otherwise, 1/n is used. "**all**" means all test data are included.

portance of event inter-dependency modeling and cross-sentence information integration for ED task.

When only information of current sentence is available, the 1-layer MLBiNet outperforms HBTNGMA by 2.9 points. It proves that the hierarchical tagging mechanism adopted by HBTNGMA is not as effective as the bidirectional decoding mechanism we proposed. Intuitively, the bidirectional decoder models event inter-dependency explicitly by a forward decoder and a backward decoder, which is more efficient than hierarchies.

3.4 Effect on Extracting Multiple Events

The existing event inter-dependency modeling methods (Chen et al., 2015, 2018; Liu et al., 2018) aim to extract multiple events jointly within a sentence. To demonstrate that sentence-level event inter-dependency modeling benefits from cross-sentence information propagation, we evaluated the performance of our model in single event extraction (1/1) and multiple events joint extraction (1/n). 1/1 means one sentence that has one event; otherwise, 1/n is used.

The experimental results are presented in Table 3. As shown, we can verify the importance of cross-sentence information propagation mechanism and bidirectional decoder in sentence-level multiple events joint extraction based on the following results: a) When only the current sentence information is available, the 1-layer MLBiNet outperforms existing methods at least by 2.4 points in 1/n case, which proves the effectiveness of bidirectional decoder we proposed; b) For ours 2-layer and 3-layer models, their performance in both 1/1 and 1/n cases surpasses the current methods by a large margin, which proves the importance of propagating information across sentences for single event and multiple events extraction. We conclude that it is the propagating information across sentences and bidirectional decoder which make cross-sentence joint event detection successful.

Methods	1-layer	2-layer	3-layer
backward	72.2	75.0	75.5
forward	72.8	76.0	76.5
bidirectional	**76.2**	**78.6**	**78.6**

Table 4: The performance of our proposed method with different multi-layer settings or decoder methods.

Methods	P	R	F_1
baseline (1-layer)	74.1	78.5	76.2
average (2-layer)	74.5	82.5	78.3
concat (2-layer)	**75.0**	82.6	**78.6**
LSTM (2-layer)	74.2	**83.7**	**78.6**

Table 5: The performance of MLBiNet with different kinds of information aggregation mechanisms.

3.5 Analysis of Decoder Layer

Table 4 presents the performance of the model in three decoder mechanisms: forward, backward and bidirectional decoder, as well as three multi-layer settings. We can reach the following conclusions: a) Under three decoder mechanisms, the performance of the proposed model will be significantly improved as the number of decoder layers increases; b) The bidirectional decoder dominates both forward decoder and backward decoder, and forward decoder dominates backward decoder; c) The information propagation across sentences will enhance event relevant signal regardless of the decoder mechanism applied. Among the three decoder models, the bidirectional decoder performs best because of its ability in capturing bidirectional event inter-dependency, which proves both the forward and backward decoders are critical for event inter-dependency modeling.

3.6 Analysis of Aggregation Model

In information aggregation module, we introduce a *LSTM* shown in (6) to aggregate sentence information, and then propagate to other sentences via the bidirectional decoder. We compare other aggregation methods: a) *concat* means the sentence information is aggregated by simply concatenating the first and last event tag vector of the sentence, and b) *average* means the sentence information is aggregated by averaging the event tag vectors of tokens in the sentence. The experimental results

are presented in Table 5.

Compared with the baseline 1-layer model, other three 2-layer settings equipped with information aggregation and cross-sentence propagation performs better. It proves that sentence information aggregation module can integrate some useful information and propagate it to other sentences through the decoder. On the other hand, the performance of *LSTM* and *concat* are comparable and stronger than *average*. Considering that the input of the information aggregation module is the event tag vector obtained by the bidirectional decoder, which has captured the sequential event information. Therefore, it is not surprising that *LSTM* does not have that great advantage over *concat* and *average*.

4 Related Work

Event detection is a well-studied task with research effort in the last decade. The existing methods (Chen et al., 2015; Nguyen and Grishman, 2015; Liu et al., 2017; Nguyen and Grishman, 2018; Deng et al., 2020; Tong et al., 2020; Lai et al., 2020; Liu et al., 2020; Li et al., 2020; Cui et al., 2020; Deng et al., 2021; Shen et al., 2021) mainly focus on sentence-level event trigger extraction, neglecting the document information. Or the document-level semantic and event inter-dependency information are modeled separately.

For the problem of event inter-dependency modeling, some methods were proposed to jointly extract triggers within a sentence. Among them, Chen et al. (2015) used dynamic multi-pooling CNN to preserve information of multiple events; Nguyen et al. (2016) utilized the bidirectional recurrent neural networks to extract events; Liu et al. (2018) introduced syntactic shortcut arcs to enhance information flow and used graph neural networks to model graph information; Chen et al. (2018) proposed a hierarchical tagging LSTM layer and tagging attention mechanism to model the event inter-dependency within a sentence. Considering that adjacent sentences also store some relevant event information, which would enhance the event signals of other sentences. These methods would miss the event inter-dependency information across sentences. For document-level event inter-dependency modeling, Lin et al. (2020) proposed to incorporate global features to capture the cross-subtask and cross-instance interactions.

The deep learning methods on document-level semantic information aggregation are primarily based on multi-level attention mechanism. Chen et al. (2018) integrated document information by introducing a multi-level attention. Zhao et al. (2018) used trigger and sentence supervised attention to aggregate information and enhance the sentence-level event detection. Zheng et al. (2019) utilized the memory network to store document level contextual information and entities. Some feature-based document level information aggregation methods were proposed by (Ji and Grishman, 2008; Liao and Grishman, 2010; Hong et al., 2011; Huang and Riloff, 2012; Reichart and Barzilay, 2012; Lu and Roth, 2012). And Zhang et al. (2020) proposed to aggregate the document-level information by latent topic modeling. The attention-based document-level information aggregation mechanisms treat all sentences in the document equally, which may introduce some noises from distant sentences. And the feature-based methods require extensive human engineering, which also greatly affects the portability of the model.

5 Conclusions

This paper presents a novel Multi-Layer Bidirectional Network (MLBiNet) to propagate document-level semantic and event inter-dependency information for event detection task. To the best of our knowledge, this is the first work to unify them in one model. Firstly, a bidirectional decoder is proposed to explicitly model the sentence-level event inter-dependency, and event relevant information within a sentence is aggregated by an information aggregation module. Then the multiple bidirectional tagging layers stacking mechanism is devised to iteratively propagate semantic and event-related information across sentence. We conducted extensive experiments on the widely-used ACE 2005 corpus, the results demonstrate the effectiveness of our model, as well as all modules we proposed.

In the future, we will extend the model to the event argument extraction task and other information extraction tasks, where the document-level semantic aggregation and object inter-dependency are critical. For example, the recently concerned document-level relation extraction (Quirk and Poon, 2017; Yao et al., 2019), which requires reading multiple sentences in a document to extract entities and infer their relations by synthesizing all information of the document. For other sequence labeling tasks, such as the named entity recognition, we can also utilize the proposed architecture

to model the entity label dependency.

Acknowledgments

We want to express gratitude to the anonymous reviewers for their hard work and kind comments. This work is funded by NSFCU19B2027/91846204, National Key R&D Program of China (Funding No.SQ2018YFC000004).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 167–176.

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1267–1276.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Shiyao Cui, Bowen Yu, Tingwen Liu, Zhenyu Zhang, Xuebin Wang, and Jinqiao Shi. 2020. Edge-enhanced graph convolution networks for event detection with syntactic relation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2329–2339.

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 151–159.

Shumin Deng, Ningyu Zhang, Luoqiu Li, Hui Chen, Huaixiao Tou, Mosha Chen, Fei Huang, and Huajun Chen. 2021. Ontoed: Low-resource event detection with ontology embedding. In *ACL*. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2019. Multi-sentence argument linking. *arXiv preprint arXiv:1911.03766*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640.

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

Lifu Huang and Heng Ji. 2020. Semi-supervised new event type induction and event detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 718–724.

Ruihong Huang and Ellen Riloff. 2012. Modeling textual cohesion for event extraction. *Proceedings of the 26th Conference on Artificial Intelligence*.

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

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Event detection: Gate diversity and syntactic importance scoresfor graph convolution neural networks. *arXiv preprint arXiv:2010.14123*.

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*, pages 829–838.

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

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

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

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)*, pages 1641–1651.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Meeting of the Association for Computational Linguistics*, pages 1789–1798.

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

Wei Lu and Dan Roth. 2012. Automatic event extraction with structured preference modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 835–844.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *International Conference on Learning Representations*, abs/1301.3781.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 365–371.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5900–5907.

Chris Quirk and Hoifung Poon. 2017. Distant supervision for relation extraction beyond the sentence boundary. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Roi Reichart and Regina Barzilay. 2012. Multi event extraction guided by global constraints. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 70–79.

Shirong Shen, Guilin Qi, Zhen Li, Sheng Bi, and Lusheng Wang. 2020. Hierarchical chinese legal event extraction via pedal attention mechanism. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 100–113.

Shirong Shen, Tongtong Wu, Guilin Qi, Yuan-Fang Li, Gholamreza Haffari, and Sheng Bi. 2021. Adaptive knowledge-enhanced bayesian meta-learning for few-shot event detection. In *Findings of ACL*. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

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

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, Volume 1 (Long and Short Papers)*, pages 998–1008.

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)*, pages 5770–5774.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. Docred: A large-scale document-level relation extraction dataset. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Wojciech Zaremba and Ilya Sutskever. 2014. Learning to execute. *arXiv preprint arXiv:1410.4615*.

Junchi Zhang, Mengchi Liu, and Yue Zhang. 2020. Topic-informed neural approach for biomedical event extraction. *Artificial Intelligence in Medicine*, 103:101783.

Tongtao Zhang, Heng Ji, and Avirup Sil. 2019. Joint entity and event extraction with generative adversarial imitation learning. *Data Intelligence*, 1(2):99–120.

Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Document embedding enhanced event detection with hierarchical and supervised attention. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 414–419.

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)*, pages 337–346.