DGMED: A Novel Document-Level Graph Convolution Network for Multi-Event Detection

Anonymous ACL submission

Abstract

Online news documents can contain thousands of characters and tens of events. To detect events in these documents, it is important to construct long-range context information. Such information, however, is not effectively created in existing event detection methods including DMBERT, MOGANED. As a result, these methods show poor event detection accuracy in production where long documents are common. To address this, this paper proposes a Document-level Graph convolution network for Multi-event Detection (DGMED). DGMED represents each sentence in a long document as a graph, and it interconnects these graphs using novel cross-sentence global neural network nodes. These nodes allow DGMED further construct accurate document-level contextual information, thus accurately extracting multiple events as required. We evaluate DGMED using a public event extraction dataset (i.e., Maven) and a production large-scale dataset (named AML). Evaluation results show that DGMED can out-perform state-of-the-art methods BERT+CRF and BiLSTM+CRF up to 0.7% in Maven and 5.7% in AML.

1 Introduction

Extracting multiple events from online news documents is an important task for NLP applications, making it one of recent popular research areas in the NLP community (Liao and Grishman, 2010; Yang et al., 2018; Zheng et al., 2019; Feng et al., 2016; Zhao et al., 2018; Nguyen and Grishman, 2015; Yang and Mitchell, 2016; Yan et al., 2019; Ma et al., 2020; Nguyen and Grishman, 2016; Chen et al., 2015a; Lai et al., 2020; Cui et al., 2020; Yang et al., 2019; Elhammadi et al., 2020). Event Detection (ED) is often implemented as a sequence labeling task (Yan et al., 2019; Cui et al., 2020; Ding et al., 2019), and it follows two steps: it 1) identifies a trigger word, and 2) assigns the trigger word to a predefined event class.

Accurately detecting multiple events in real-world news documents is however challenging. These documents contain thousands of characters and describe tens of events. To detect events in these documents, we find out that *long-range contextual modeling* must be implemented in ED methods; otherwise, the accuracy of these methods can largely suffer. We illustrate this using Figure 1. In the first fragment, the 2nd event is detected based on a trigger word "*fined*". The 1st event is detected based on "*released*" which follows a subject "*China Banking and Insurance Regulatory Commission (CBIRC)*". Since CBIRC is a financial regulatory department, the 2nd event can be thus inferred as an "*anti-money laundering Regulatory Penalty event*". In the second fragment, 2nd event can be still detected again based on "*fined*"; however, the 1st event is detected based on "*punished*" which follows the subject "*Municipal Supervision Bureau (MSB)*". Since MSB is not a financial regulatory department, the 2nd event is thus a "*non-anti-money laundering event*". This event type is different from the one detected in the first fragment, and such a detection error can be handled using effective long-range contextual modelling.

Though important, long-range contextual modelling is still poorly implemented in existing ED methods. Domain-specific ED methods (Yang et al., 2018; Zheng et al., 2019; Xu et al., 2021) use heuristics to detect events, and they show poor accuracy in practice. Neural-network-based ED methods, such as k-gram-based CNNs (Nguyen and Grishman, 2015, 2016), use hierarchical attention-based models (Chen et al., 2018) to capture contextual dependencies; but they fail to identify long-range syntactic dependencies. More recent ED methods address this using syntactic models (Liu et al., 2018; Buyko et al., 2009), such as syntactic Graph Convolutional Networks (GCNs) (Nguyen and Grishman, 2018), syntactic transformers (Ma et al., 2020) and graph attention networks (Yan

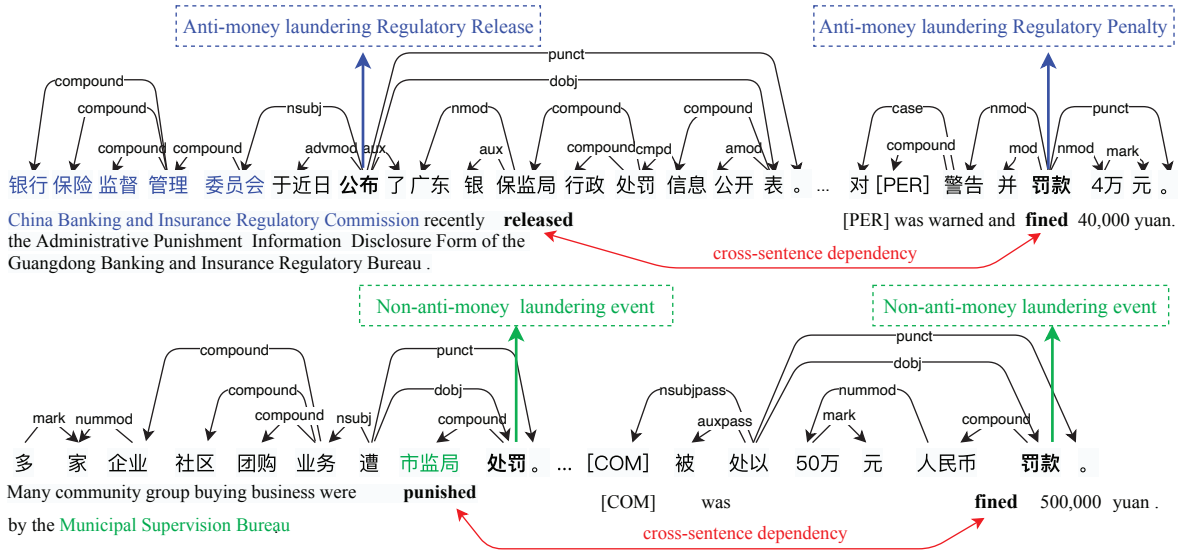

Figure 1: Documents fragments that reflect the importance of long-range contextual modelling. Sensitive data are masked using wildcards [PER] and [COM].

et al., 2019). These models, however, suffer from syntactic parsing errors, and these errors often propagate through entire neural networks, making these models difficult to be used by accuracy-sensitive applications.

In this paper, we propose a **D**ocument-level **G**raph convolution network for **M**ulti-**E**vent **D**etection (DGMED). This method realizes the anticipated long-range contextual modelling. The design of DGMED addresses several challenges in processing long news documents: (1) contextual words which can identify event types are often scattered across multiple sentences; (2) multiple events often share overlapping contexts. To address these challenges, DGMED divide a document-level graph into several graphs: each graph corresponds to a sentence in a document. DGMED further carefully construct *global nodes* to connect the sentence-level graphs, and create document-level contextual information sharing through passing messages among sentence-level contexts.

Further, we design a large-scale multi-event extraction dataset to evaluate DGMED. This dataset, named Anti-Money Laundering (AML), contains 3,924 financial news documents collected from real-world websites. These documents contain a large number of events which must be extracted to support numerous downstream NLP applications in our production. We employ annotators to annotate these documents. These documents exhibit a high multi-event ratio of 93%, substantially higher than existing ED datasets, e.g., ACE 2005 (Walker et al., 2006) and KBP 2015 (Ellis et al., 2015).

Evaluation results show that DGMED not only out-perform State-Of-The-Art (SOTA) methods (i.e., BiLSTM+CRF) on the large-scale AML dataset by up to 5.7%. It also out-performs SOTA methods (i.e., BERT+CRF) on a public ED dataset: MAVEN (Wang et al., 2020), showing the effectiveness and generality of DGMED.

2 Related Work

This section describes the related work of DGMED. Event detection is an important sub-task of Event Extraction (EE). Early studies use manually generated features, such as lexical, syntactical or contextual features (Yang and Mitchell, 2016). Manual features often lack contextual information that is rather important for accurately detecting events. Recent studies thus used deep neural networks, e.g., Convolutional Neural Networks (CNNs) and Long Short Term Memory Networks (LSTMs), for modelling contextual information (Yang et al., 2018; Feng et al., 2016; Zhao et al., 2018; Nguyen and Grishman, 2015; Zheng et al., 2019; Nguyen and Grishman, 2016; Chen et al., 2015a; Liu et al., 2018; Ding et al., 2019; Chen et al., 2018; Duan et al., 2017). For example, DCFEE (Yang et al., 2018) uses a BiLSTM-CRF model to learn features of financial events, the Doc2EDAG model (Zheng et al., 2019) learns a neural embedding for entities, sentences and documents, and the BERT-MLP model (Yang et al., 2019) uses a pre-trained BERT to encode sentences. (Deng et al., 2021)

proposed ontology-based model to handle new unseen event types. (Pouran Ben Veyseh et al., 2021) use pre-trained language model GPT-2 to generate training samples for ED. MLBiNet (Lou et al., 2021) reformulate ED as a seq2Seq task to model document-level contexts and event relations.

More recently, graph models have attracted much attention in natural language processing (Yao et al., 2019). There are several studies that attempted to implement ED with graph models, and they achieved better performance compared to above neural network counterparts (Yao et al., 2019; Yan et al., 2019; Nguyen and Grishman, 2018; Lai et al., 2020; Cui et al., 2020). However, existing graph models often focus on short-document scenario, and they only build sentence-level syntactic dependency trees. Although these models can further improve their performance by using syntactic rules, multi-skip dependency, gated convolution, or rebalancing data distribution (Cao et al., 2020; Tong et al., 2020; Wang et al., 2019a; Huang and Ji, 2020), they still exhibit insufficient performance in processing long financial news documents with multiple events. This makes it necessary to further explore new GCN designs that can effectively implement multi-event extraction in long documents, which is the focus of this paper.

3 Method

In this section, we introduce the design of DGMED. Figure 2 presents an overview of DGMED. The input of DGMED is a document. This document is encoded (by Feature extractor), and then passed to a syntactic-aware-GCN layer which creates a graph that describes the information in each sentence. Multiple sentence-level graphs are connected using global nodes. These nodes are passed to a CRF layer (Lafferty et al., 2001b) where multiple events are eventually detected. In the following, we will describe the designs of these layers in details. Throughout our description, we use $D = \{s_i\}_1^m$ to denote a document, $s_i = \{w_{ij}\}_1^n$ to denote a sentence, where s_i is the i-th sentence and w_{ij} is the j-th token in i-th sentence.

3.1 Encoder Layers

To support event extration, DGMED must first encode a given document. Given that most of the documents in our production are English and Chinese, we implement two encoder layers in DGMED: BiLSTM (Hochreiter and Schmidhuber, 1997) and

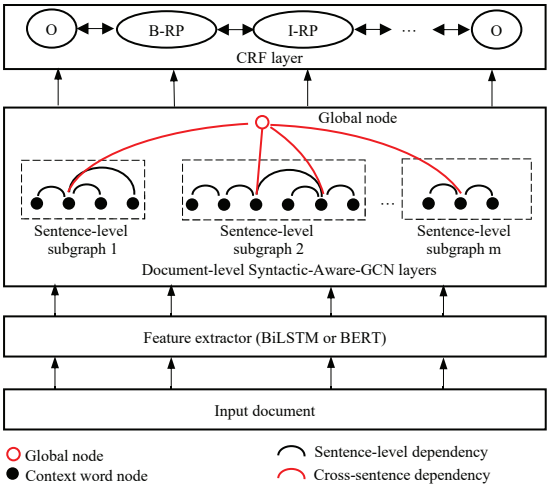

Figure 2: The overview of DGMED.

BERT (Devlin et al., 2018). The BiLSTM encoder is used to encode Chinese corpus, and BERT is used to encode English corpus. The BiLSTM encoder concatenate word embedding $word_i$, entity type embedding et_i, position embedding ps_i and POS tag embedding pos_i to build word embedding. The BERT encoder sums token embedding, segment embedding and position embedding as input.

3.2 Document-level GCN Layers

Inspired by prior GCN-based ED methods, we build a graph for each sentence and represent each word as a graph node, and the link between two words in a dependency tree as an edge. The graph is represented by an $n \times n$ adjacency matrix A_i, where n is the total number of words in the i-th sentence.

We follow classic GCNs (Kipf and Welling, 2017) which use a scalable approach for semi-supervised learning on graph data. Considering an L-layer GCN where $l \in [1, ..., L]$, if H_i^{l-1} denotes the input state and H_i^l denotes the output state of the i-th sentence of the l-th layer. This GCN can be formally defined as:

$$\begin{aligned} H_i^l &= GCN(A_i, H_i^{l-1}, W) \\ &= \sigma(A_i H_i^{l-1} W) \end{aligned} \quad (1)$$

where $A_i \in R^{n \times n}$ is the adjacency matrix, W is a trainable filter matrix and σ is a nonlinear activation function, e.g., ReLU.

We notice that the dependency relations between words are not equally important. Motivated by (Nguyen and Grishman, 2018), We multiply A_i by a weighted edge matrix V, in which each element

V_{xy} represents the weight of the edge between node x and node y, to distinguish between dependency relations. Finally, our convolutional operation can be defined as:

$$H_i^l = \sigma(A_i \circ V H_i^{l-1} W) \quad (2)$$

where V_{xy} in V is obtained by looking up a one-dimensional p-length vector parameter, p is the total number of all possible relations between nodes. In the following, we denote this method as $S\text{-}GCN$.

Modeling document-level context. By far we have embeddings for all words after the syntactic-aware GCN layer; but these embeddings encode contextual information only at the sentence level. To build cross-sentence context (i.e., document-level), we propose to construct global nodes which allow information to be exchanged among sentences.

As shown in Figure 2, the global nodes are connected with candidate trigger words to gather cross-sentence information. We treat all verbs as candidate trigger words, and global nodes are not connected with all word nodes, which avoid incurring excessive noise on the graph. Global nodes can be regarded as virtual hubs to gather and propagate information from and to word nodes. We initialize the embeddings of global nodes randomly.

Memory-efficient alternate update strategy. We need to train the document-level graph that comprises of sentence-level graphs connected by global nodes. A key challenge is that the document-level graph is often large in size, and it is difficult to be fitted into the memory of a GPU. For example, given a document that has 100×128 words after padding (here, 100 is the number of sentences and 128 is the number of words of each sentence). To process such a document, we would need to create an adjacency matrix in the size of $(12800 + G)^2$ where G is the number of global nodes to create. To process such a matrix efficiently, we propose a memory-efficient alternate update (i.e., training) strategy. This strategy divides the update process of l-th layer into two phases: 1) updating global nodes and 2) updating word nodes for each sentence.

Document-level Syntactic-Aware-GCN layers in Figure 2 shows the process of the memory-efficient alternate update strategy. At the first phase, we focus on updating the global nodes. This can be achieved by constructing a sub-graph that contains the global nodes as well as their neighbors (i.e., candidate trigger word nodes and their associated edges). As the red part shows, this sub-graph is a bipartite graph and it is used for updating global nodes and learn document-level knowledge.

We then formally define the update strategy. Global node embeddings of the l-th layer G^l can be updated based on the following formula:

$$\left[\_; G^l\right] = S - GCN(A_d, \left[H_{trigger}^{l-1}; G^{l-1}\right], W_d) \quad (3)$$

where A_d represents the adjacency matrix of trigger candidate nodes and global nodes, and $H_{trigger}^{l-1}$ represents candidate trigger word node embeddings in the $l-1$-th layer. W_d is a learnable parameter.

At the second phase, each sentence-level sub-graph is connected to trained global nodes respectively. Word embeddings within the sentence can be therefore updated with both local and document-level information. As shown in Figure 2, each update step only requires a sub-graph with global nodes and word nodes from a single sentence.

The embeddings of the i-th sentence H_i can be formulated as follows:

$$\left[H_i^l; \_\right] = S - GCN(\tilde{A}_i, \left[H_i^{l-1}; G^l\right], W_s) \quad (4)$$

$$H^l = \left[H_1^l; H_2^l; ...; H_m^l\right] \quad (5)$$

where H^l represents the document word embeddings of the l-th layer, and $\tilde{A}_i$ represents sentence-level graph adjacency matrix of the i-th sentence and global nodes. W_s is a trainable parameter shared between sentences.

By alternately updating global nodes and word nodes, DGMED can capture all sentence-level and document-level information without consuming tremendous GPU memory. By contrast, BiLSTM-based DGMED of two phases contains 25M parameters compared to 187M of one phase model. For BERT-based DGMED, the parameters of the two models are 280M and 118M, respectively. This alternate update process is formally defined in Algorithm 1.

3.3 CRF Layer

After aggregating the node representations of syntactic-aware GCN layers, we build a fully-connected network to project the final hidden output state h:

$$p(y_t|h) = softmax(W_t h + b_t) \quad (6)$$

Algorithm 1 The alternate update strategy

Input: Number of layers L, embeddings of tokens e_1, ... , e_n, initialized embedding of global nodes g_1, ... , g_k, end positions of sentences p_1, ... , p_m, positions of candidate trigger words t_1, ... , $t_{|triggers|}$.

Output: Updated embeddings of tokens e_1, ... , e_n.

```
//initialization
G^0 ← [g_1; ...; g_k]
for l from 1 to L do
    H_1^{l-1} = [e_0; e_1; ...; e_{p_1}]
    ...
    H_m^{l-1} = [e_{p_{m-1}+1}; e_{p_{m-1}+2}; ...; e_{p_m}]
    H_trigger^{l-1} = [e_{t_1}; e_{t_2}...; e_{t_|triggers|}]
    //update global nodes
    [_; G^l] =S-GCN(A_d, [H_trigger^{l-1}; G^{l-1}])
    //update word nodes of each sentence
    for i from 1 to m do
        [H_i^l; _] =S-GCN(Ã_i, [H_i^{l-1}; G^l])
        [e_{p_{i-1}+1}; ...; e_{p_i}] = H_i^l
    end for
end for
```

NO.EVT	NO.DOC	PROP(%)
1	219	7.08
[2, 10]	1,565	50.60
[11, 20]	865	27.97
[21,)	444	14.35

Table 1: Statistics of documents and associated events.

Event Type	NO.ANN	NO.DOC
RP	26,990	2,751
RR	6,221	2,400
RI	2,054	1,115
JC	703	384
RV	159	79
Total	36,127	6,729

Table 2: Counts of annotations and event types.

Dataset	Domain	Label	Size	MER
ACE2005	general	manual	599	N/A
MAVEN	general	manual	3,623	100%
DCFEE	finance	auto	2,976	3%
AML	finance	manual	3,924	93%

Table 3: Dataset comparison.

where y_t is the tag label sequence, W_t maps the word representation h to the feature score for each event type and b_t is a bias term.

It has been shown crucial to handle the priori transition probabilities between labels in sequence labeling. This is however not considered in previous graph-based event detection models. To close this gap, we propose to place a CRF layer after the fully-connected network. Let J denote the number of all possible transition paths of labels, we adopt the negative log-likelihood loss function as our optimization objective:

$$loss = -log(\frac{e^{S_j}}{\sum_{j=1}^{J} e^{S_j}}) \quad (7)$$

here,

$$S_j = \sum_{t=1}^{N} \phi_j p(y_t|h) + \sum_{t=2}^{N} \psi_j p(y_{t-1}, y_t|h) \quad (8)$$

where ϕ_j and ψ_j are the emission score function and transition score function, respectively.

4 AML Dataset

In this section, we describe the design and statistics of the AML dataset. In production, we support a large number of financial applications which need to automatically detect events relevant to anti-money laundering regulations. To help design methods to detect these events, we initially collected more than 8,000 financial news documents from widely-used Chinese financial websites, including China Economic Information Networks (CEIN, 2021) and Sina Finance (Sina Corporation, 2021). These documents were published between 2018 and 2020. After cleaning the collected documents, the dataset eventually contains 3,924 documents. These documents comprise of 1,485 characters on average, ranging from 21 characters to 5,113 characters.

Statistics of events and annotation. The documents in the AML dataset have 5 event types: *Regulatory Penalty (RP)*, *Regulatory Release (RR)*, *Regulatory Investigation (RI)*, *Judicial Case (JC)*, *Regulatory View (RV)*. We employed 5 professional annotators to label trigger words by the most relevant event types following the "BIO" annotation scheme. Since each event type has 2 particular labels "B" and "I" and all event types share the same label "O", the total number of tags needed is $2P + 1$, where P is the number of predefined event types. Each sample is annotated by two annotators. If their annotation results are different, an

Method	AML			MAVEN		
	P	R	F1	P	R	F1
DMCNN	70.3±0.0	67.4±0.5	68.8±0.1	66.3±0.9	55.9±0.5	60.6±0.2
BiLSTM	77.2±0.1	72.8±0.5	74.9±0.2	59.8±0.8	67.0±0.8	62.8±0.8
BiLSTM+CRF	77.6±0.2	75.5±0.1	76.5±0.2	63.4±0.7	64.8±0.7	64.1±0.1
MOGANED	79.4±0.4	80.6±0.3	80.0±0.3	63.4±0.9	64.1±0.9	63.8±0.2
DMBERT	**81.5±0.5**	80.1±0.1	80.8±0.2	62.7±1.0	**72.3±1.0**	67.1±0.4
BERT+CRF	81.0±0.3	81.6±0.2	81.3±0.2	65.0±0.8	70.9±0.9	67.8±0.2
DGMED(BiLSTM)	81.5 ±0.3	**82.9±0.1**	**82.2±0.1**	63.7 ±0.1	67.9 ±0.4	65.7 ±0.2
DGMED(BERT)	–	–	–	**65.8±0.2**	71.3 ±0.3	**68.5±0.2**

Table 4: The overall trigger classification performance of various models on AML and MAVEN.

extra annotator is employed until the difference is resolved.

Table 1 presents a summary of events in the documents, where NO.EVT denotes the number of events that a document contains, NO.DOC denotes the number of documents that correspond to a certain range of event counts, and PROP(%) denotes the proportion of corresponding documents. As we can see, up to 93% of documents contain more than 2 events, and over 40% of documents contain more than 11 events.

Table 2 further provides an in-depth analysis of event types and associated annotation in the AML dataset. NO.ANN is the number of trigger words for an event type. NO.DOC is the number of documents in which an event type occurs. As we can see, the documents contain a balanced distribution of event types, and there are sufficient annotations for each event type.

We also examine the quality of annotation. To this end, we randomly selected 200 documents and invited a NLP expert to annotate these documents independently. We regard this NLP expert's annotation as ground-truth. The precision is 97.6%, and the recall is 96.9%, implying the high quality of annotation in the AML dataset.

Dataset comparison Table 3 compares the AML dataset with other widely used ED datasets: ACE 2005 (Walker et al., 2006), MAVEN (Wang et al., 2020), DCFEE (Yang et al., 2018). We compare these datasets in four aspects: data domains, labeling methods, dataset sizes and multi-event ratios. ACE 2005 contains 1,800 manually labeled documents, but it has only 599 Chinese documents. Similarly, MAVEN contains 4,480 manually labeled documents in total including 3,623 publicly available train and development set, but all in English. The multi-event ratio(MER) for DCFEE is only 3%, which is far not enough for building multi-event detection models. The Doc2EDAG (Zheng et al., 2019) dataset does not contain trigger words which are important for detecting events. Thus, it can be only used for event argument extraction. Compared to all these datasets, AML exhibits a high multi-event ratio: 93% and it contains the largest collection of financial documents with high-quality manual labels.

5 Experiments

In this section, we evaluate the DGMED method and compare it with SOTA methods on the AML dataset and the MAVEN dataset (Wang et al., 2020). MAVEN is a general English event detection dataset with 168 event types. (Yu et al., 2021) propose a lifelong learning framework for event detection on MAVEN. However, they only evaluate their model on the development set. For the AML dataset, we randomly selected 80% documents in the AML dataset for training, 10% for validation, and 10% for testing. The MAVEN dataset contains 2913 training samples, 710 validation samples, and 857 test samples. We submit the predictions of DGMED to a competition hosted on CodaLab (CodaLab, 2020). We adopt Precision (P), Recall (R) and F-measure (F1) as main evaluation metrics.

We use the Stanford Chinese CoreNLP toolkit (Stanford NLP Group, 2021) for sentence splitting, tokenizing, named entity recognition (NER), POS-tagging and dependency parsing. We obtain a pre-trained word embedding by using fastText algorithm (Joulin et al., 2017) on the Baidu Tieba Chinese corpus (Baidu Corporation, 2021).

We run experiments on a NVIDIA Tesla P100

Model	RP			RR			RI			JC			RV		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
DMCNN	72.5	69.8	71.1	61.2	66.0	63.5	64.8	73.1	68.7	0.0	0.0	0.0	0.0	0.0	0.0
BiLSTM	80.5	75.0	77.7	65.7	71.8	68.6	72.2	80.1	75.9	0.0	0.0	0.0	0.0	0.0	0.0
BiLSTM+CRF	80.8	78.3	79.5	68.3	71.9	70.1	70.8	79.8	75.0	38.7	12.1	18.5	**40.0**	9.1	14.8
MOGANED	**88.3**	76.4	81.9	66.8	82.0	73.6	69.1	**86.4**	76.8	29.7	27.4	28.5	0.0	0.0	0.0
DMBERT	83.7	81.1	82.4	69.4	80.4	74.5	75.8	83.3	79.4	35.5	28.8	31.8	26.2	56.2	35.7
BERT+CRF	82.7	80.4	81.5	70.4	**83.6**	76.4	73.5	84.5	78.6	29.7	20.0	23.2	0.0	0.0	0.0
DGMED(BiLSTM)	83.1	**84.7**	**83.9**	**73.7**	82.0	**77.6**	**83.8**	84.1	**83.9**	**41.0**	**37.3**	**39.1**	35.7	**62.5**	**45.5**

Table 5: Results on the AML dataset. Event-level precision (P), recall (R) and F1-score evaluated on the test set.

GPU. For BiLSTM, we use the same embedding size of 50 for word embedding, entity type embedding, position embedding and POS tagging embedding. In a downstream neural network, we enlarge the hidden units of BiLSTM encoder and syntactic-aware-GCN layer to 200 and 128, respectively. We adopt batch size as 32, the learning rate as 0.001, and the number of global nodes in each layer as 2. For BERT, we use BERT_{base} (Devlin et al., 2018) as the feature extractor. The model checkpoints and implementation are from MAVEN.

We compare DGMED with:

1. DMCNN (Chen et al., 2015b) is a CNN-based model for extracting events.
2. BiLSTM (Hochreiter and Schmidhuber, 1997) uses forward LSTM and a backwards LSTM to extract events.
3. MOGANED (Yan et al., 2019) is a GCN that uses aggregated attention to model multi-level syntactic representations in a sentence.
4. DMBERT (Wang et al., 2019b) is a BERT-based model which uses a dynamic multi-pooling layer to aggregate features.
5. BiLSTM+CRF and BERT+CRF use CRF (Lafferty et al., 2001a) as output layers, and use BiLSTM and BERT as feature extractors, respectively.

5.1 Overall Performance

Table 4 shows the performance results of DGMED and baselines. Considering that most Chinese BERT models are built at the character level and dependency parsing relations for GCN are built at the token level, we didn't conduct the DGMED(BERT), i.e. BERT+DOC-GCN+CRF, experiment for AML dataset because this leads to inconsistencies of different layers of the model. Experimental results on both the two datasets show that DGMED model outperforms all other baselines. DGMED(BiLSTM), i.e. BiLSTM+DOC-GCN+CRF, get 5.7% and 1.6% promotion on F1-score compared with BiLSTM+CRF on two datasets respectively. And DGMED(BERT) outperforms BERT+CRF by 0.7% on MAVEN dataset. Table 5 further shows that F1-scores based on event types of AML dataset, DGMED achieves the best overall performance. The significant improvement in F1-score demonstrates the importance of implementing document-level graph construction.

5.2 Parameter Study

We then evaluate DGMED with different parameter settings and the indispensability of each key module on AML dataset.

Long-text scenarios. The global nodes in DGMED are effective in modelling long-range contextual information, especially in long text. To show this, we build a long-text-centric testing dataset which contains only the documents with lengths over 800 characters. As shown in Table 6, DGMED(BiLSTM)'s F1-score out-performs baselines by 1.7∼15.6. Moreover, its F1-decrease is less than all other models including BERT-based and BiLSTM-based models. This demonstrates that the use of DOC-GCN is the key to making models work in long-text scenarios.

Number of global nodes. We then evaluate the impact of the number of global nodes (denoted by No.GN in Table 8). When No.GN is 1, DGMED achieves comparable performance with other baselines. When increasing No.GN to 2, DGMED starts to out-performing baselines and it achieves the highest F1-score. An interesting observation

Model	P	R	F1
DMCNN	67.3	64.2	65.7(↓ 3.1)
BiLSTM	73.8	70.5	72.1(↓ 2.8)
BiLSTM+CRF	76.7	71.3	73.9(↓ 2.6)
MOGANED	77.9	79.1	78.5(↓ 1.5)
DMBERT	79.4	78.6	79.0(↓ 1.8)
BERT+CRF	79.3	79.9	79.6(↓ 1.7)
DGMED(BiLSTM)	80.4	82.3	81.3(↓ **0.8)**

Table 6: Results on the long-text datasets.

Model	P	R	F1
DGMED(BiLSTM)	81.5	82.9	**82.2**
-document-level graph	81.8	80.2	81.0
-document-level graph -CRF layer	82.5	79.2	80.8
-document-level graph -syntactic-aware method	75.7	85.0	80.1
-document-level graph -syntactic-aware method -CRF layer	76.3	82.3	79.2

Table 7: Ablation study results on the AML dataset.

NO. GN	P	R	F1
1	80.3	**84.0**	82.1
2	**81.5**	82.9	**82.2**
3	81.0	82.9	81.9
4	80.6	82.8	81.7

Table 8: Performance of CFMED with different numbers of global nodes. NO.GN refers to number of global nodes.

is that the performance of DGMED does not always increase with more global nodes. A possible explanation is that: if we use an excessive number of global nodes, the document-level graph in DGMED will end with a large number of unnecessary edges. These edges can result in extra noises, which can adversely affect the performance. With a few global nodes, we cannot identify complex cross-sentence dependencies. The optimal number of global nodes depends on the types of documents, and we are working towards automatically choosing this number.

Ablation study. DGMED has three novel components: (1) a document-level graph to learn the information across sentences using global nodes, (2) a syntactic-aware GCN layer to distinguish dependency relations, (3) a CRF layer to handle priori transition probabilities between labels. To evaluate the performance gain by each component, we will remove these components in DGMED sequentially, and show their performance results in Table 7. We first remove the document-level graph, and the F1-score drops by 1.2%. We then remove the CRF layer and the syntactic-aware method in order, and the F1-score drops by 1.4% and 2.1%, respectively. Finally, if we remove all the components, the F1-score of DGMED will drop by 3.0%. These results show that all the novel components in DGMED can contribute to the performance improvement in F1-score.

6 Conclusion

This paper introduces DGMED, a novel method that can effectively extract multiple events from long documents. DGMED contains novel syntactic-aware GCN layers which can filter out irrelevant syntactic neighbors, thus improving event detection accuracy. It also contains novel global nodes which can connect sentence-level graphs, thus creating required long-range contextual information. We create a new dataset AML which contains massive long documents associated with multiple important events. AML contains high-quality data annotation and it is suitable to evaluate multi-event extraction at scale. Experimental results show that DGMED can out-perform SOTA methods on both public ED datasets and the AML dataset.

References

Baidu Corporation. 2021. Baidu tieba. https://tieba.baidu.com/index.html. Accessed: 2021-09-09.

Ekaterina Buyko, Erik Faessler, Joachim Wermter, and Udo Hahn. 2009. Event extraction from trimmed dependency graphs. In *BioNLP@HLT-NAACL*, pages 19–27. Association for Computational Linguistics.

Pengfei Cao, Yubo Chen, Jun Zhao, and Taifeng Wang. 2020. Incremental event detection via knowledge consolidation networks. In *EMNLP*, pages 707–717. Association for Computational Linguistics.

CEIN. 2021. China economic information networks. https://www.cei.cn/. Accessed: 2021-09-09.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015a. Event extraction via dynamic multi-pooling convolutional neural networks. In *IJCNLP*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015b. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL-IJCNLP*, pages 167–176, Beijing, China. Association for Computational Linguistics.

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 *EMNLP*, pages 1267–1276, Brussels, Belgium. Association for Computational Linguistics.

CodaLab. 2020. Maven event detection challenge. `https://competitions.codalab.org/competitions/27320#results`. Accessed: 2021-11-15.

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 *Finds of EMNLP*, pages 2329–2339. Association for Computational Linguistics.

Shumin Deng, Ningyu Zhang, Luoqiu Li, Chen Hui, Tou Huaixiao, Mosha Chen, Fei Huang, and Huajun Chen. 2021. OntoED: Low-resource event detection with ontology embedding. In *ACL-IJCNLP*, pages 2828–2839, Online. 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. *CoRR*, abs/1810.04805.

Ning Ding, Ziran Li, Zhiyuan Liu, Haitao Zheng, and Zibo Lin. 2019. Event detection with trigger-aware lattice neural network. In *EMNLP-IJCNLP*, pages 347–356. Association for Computational Linguistics.

Shaoyang Duan, Ruifang He, and Wenli Zhao. 2017. Exploiting document level information to improve event detection via recurrent neural networks. In *IJCNLP*, pages 352–361, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Sarah Elhammadi, Laks V. S. Lakshmanan, Raymond T. Ng, Michael Simpson, Baoxing Huai, Zhefeng Wang, and Lanjun Wang. 2020. A high precision pipeline for financial knowledge graph construction. In *COLING*, pages 967–977. International Committee on Computational Linguistics.

Joe Ellis, Jeremy Getman, Dana Fore, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie M. Strassel. 2015. Overview of linguistic resources for the TAC KBP 2015 evaluations: Methodologies and results. In *TAC*. NIST.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Lifu Huang and Heng Ji. 2020. Semi-supervised new event type induction and event detection. In *EMNLP*, pages 718–724, Online. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *EACL*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*. OpenReview.net.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001a. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001b. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289. Morgan Kaufmann.

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *EMNLP*, pages 5405–5411, Online. Association for Computational Linguistics.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *ACL*, pages 789–797, Uppsala, Sweden. Association for Computational Linguistics.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *EMNLP*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

Dongfang Lou, Zhilin Liao, Shumin Deng, Ningyu Zhang, and Huajun Chen. 2021. MLBiNet: A cross-sentence collective event detection network. pages 4829–4839, Online. Association for Computational Linguistics.

Jie Ma, Shuai Wang, Rishita Anubhai, Miguel Ballesteros, and Yaser Al-Onaizan. 2020. Resource-enhanced neural model for event argument extraction. In *Findings of EMNLP*, pages 3554–3559, Online. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *ACL*, pages 365–371. The Association for Computer Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2016. Modeling skip-grams for event detection with convolutional neural networks. In *EMNLP*, pages 886–891, Austin, Texas. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *AAAI-IAAI-EAAI*, pages 5900–5907. AAAI Press.

Amir Pouran Ben Veyseh, Viet Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2021. Unleash GPT-2 power for event detection. pages 6271–6282, Online. Association for Computational Linguistics.

Sina Corporation. 2021. Sina finance. https://finance.sina.com.cn/. Accessed: 2021-09-09.

Stanford NLP Group. 2021. Stanford corenlp. https://nlp.stanford.edu/software/stanford-dependencies.html. Accessed: 2021-09-09.

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 *ACL*, pages 5887–5897, Online. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus ldc2006t06. In *Web Download. Philadelphia: Linguistic Data Consortium, 2006.*

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019a. Adversarial training for weakly supervised event detection. In *NAACL*, pages 998–1008, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019b. Adversarial training for weakly supervised event detection. In *NAACL*, pages 998–1008, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A Massive General Domain Event Detection Dataset. In *EMNLP*, pages 1652–1671, Online. Association for Computational Linguistics.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 2021. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In *ACL*. The Association for Computer Linguistics.

Haoran Yan, Xiaolong Jin, Xiangbin Meng, Jiafeng Guo, and Xueqi Cheng. 2019. Event detection with multi-order graph convolution and aggregated attention. In *EMNLP*, pages 5765–5769. Association for Computational Linguistics.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *NAACL-HLT*, pages 289–299. The Association for Computational Linguistics.

Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, and Jun Zhao. 2018. DCFEE: A document-level chinese financial event extraction system based on automatically labeled training data. In *ACL*, pages 50–55. Association for Computational Linguistics.

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

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *AAAI*, pages 7370–7377. AAAI Press.

Pengfei Yu, Heng Ji, and Prem Natarajan. 2021. Lifelong event detection with knowledge transfer. pages 5278–5290, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Document embedding enhanced event detection with hierarchical and supervised attention. In *ACL*, pages 414–419. Association for Computational Linguistics.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2edag: An end-to-end document-level framework for chinese financial event extraction. In *EMNLP-IJCNLP*, pages 337–346. Association for Computational Linguistics.