International Journal of Innovative Computing, Information and Control
Volume **19**, Number **1**, February **2023**

ICIC International ©2023 ISSN 1349-4198
pp. **47–60**

SYNED: A SYNTAX-BASED LOW-RESOURCE EVENT DETECTION METHOD FOR NEW EVENT TYPES

Ruiliu Fu[1,2], Han Wang[1,2], Xuejun Zhang[1,2], Jun Zhou[1] and Yonghong Yan[1,2,*]

[1]Speech and Intelligent Information Processing Lab
Institute of Acoustics, Chinese Academy of Sciences
No. 21, North 4th Ring Road, Haidian District, Beijing 100190, P. R. China
{ furuiliu; wanghan; zhangxuejun; zhoujun }@hccl.ioa.ac.cn
*Corresponding author: yanyonghong@hccl.ioa.ac.cn

[2]School of Electronic, Electrical and Communication Engineering
University of Chinese Academy of Sciences
No. 19(A), Yuquan Road, Shijingshan District, Beijing 100049, P. R. China

Received May 2022; revised September 2022

Abstract. *Event detection (ED) is an important and challenging information extraction task, which aims to identify triggers from unstructured text and classify them into an event type. Most of the current supervised ED methods rely heavily on high-quality annotation data and are difficult to use with new types. In order to solve the event detection task of new types in low-resource scenarios, we proposed SynED, a low-resource ED system based on syntactic embedding. We first obtain syntactic information through natural language processing (NLP) tools and combine them into syntactic embeddings which are then sent to a syntactic-gated Transformer model to extract event triggers. To create the appropriate event type ontology, we combine the syntactic and textual embeddings of event triggers. Through the correlation of event types, we realize the knowledge transfer from the seen event type ontology to the new event type ontology. Experimental results on the ACE 2005 and MAVEN datasets show that the SynED model based on syntactic embedding achieves state-of-the-art performance for new event types without annotation. Compared with the previous low-resource ED methods, our proposed SynED model has more overt benefits in the case of fewer training data and has a greater capacity for application expansion. We also explored the role of syntactic and textual embeddings in low-resource ED, demonstrating the significance of introducing syntactic information.*
Keywords: Event detection, Syntactics, Low-resource, Natural language processing, Knowledge transfer

1. **Introduction.** Event detection (ED) is a task of automatically extracting event structural information (specifically, event triggers and types) from unstructured text [1]. For example, in the event mention "*The battle was fought at the Silesian town*", an ED model is expected to extract the trigger "*Fought*" and classify it as "*Conflict*". The structural event information can promote various applications, such as biological sciences [2, 3], financial analysis [4, 5], and fake news detection [6, 7].

Traditional ED studies [1, 8-20] mainly focus on supervised learning methods and depend heavily on manual annotation, making it challenging to adapt to new event types without extra annotation effort [21]. Real-world applications expect that ED methods can be flexibly adapted to new event types in low-resource scenarios [22].

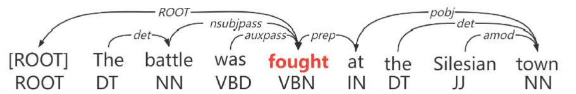
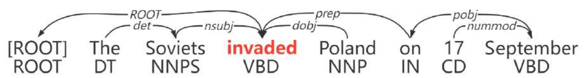
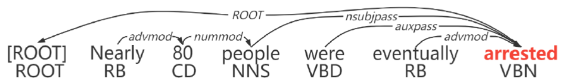
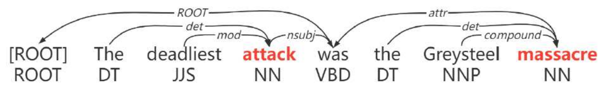
DOI: 10.24507/ijicic.19.01.47

Recently, several efforts have explored low-resource ED approaches for new event types. Huang et al. [21, 30] proposed to learn event ontology representations for each seen event type and new event type from event mentions, which can be trained using a limited amount of seen type data and expanded to new event types. Deng et al. [22] further considered the inter-structures of event types and expanded the event ontology by modeling the internal correlation between seen event types and new event types.

Although substantial advancements have been achieved, the existing low-resource ED methods are still struggling in event types with scant or no training data, which severely restricts their practical application. We analyze that this is because the generic characteristics of different event domains are not actually captured, and the acquisition of the cross-domain detection ability of the existing methods still relies on the modeling of the data distribution in the new domain. Instead, we believe that designing generic features across domains is more beneficial to enhancing the capacity to accommodate new event types, which has received scant attention in prior research. We are committed to building a generic feature and low-resource ED model that can expand effectively to new event types even in the absence of pertinent training data.

As a generic language feature, syntactic information has been shown to play an important role in ED [11, 34]. Table 1 displays the parsing results of some event mentions as obtained by the SpaCy library, a well-known NLP tool[1]. We can see that event information presents obvious regularity in the results of syntactic analysis, which is helpful for event detection. Event triggers, for instance, are frequently employed as the primary word in dependency analysis, and typically appear in generic verbs or nouns in part of speech (POS) analysis. For new event types without annotation, textual information cannot effectively discriminate but rather confuses the model, while syntactic information is almost universal for different event types and the syntactic knowledge learned from existing event types can be effectively migrated to new event types. Therefore, we propose to introduce syntactic embedding to improve the detection ability of new event types under low-resource conditions.

Table 1. Examples of parsing result of some event mentions, where bold-faced words represent the event triggers

Event mention	Event types
	Conflict
	Attack
	Arrest
	Attack killing

In this paper, we construct a syntactic embedding and propose SynED, a novel low-resource ED approach with syntactic embedding. We first build the syntactic embedding with some syntactic information obtained by the SpaCy library. The syntactic embedding is then encoded to predict event triggers using a syntactic-gated Transformer model. We combine the syntactic and textual embeddings of event triggers to construct the

[1]https://spacy.io

corresponding event type ontology and enrich the ontology of new event types through the correlation between event types. Finally, we achieve the best trigger classification result based on the similarity between the trigger embedding and the ontology of each event type.

Our contributions can be summarized as follows.

- We propose a novel SynED model based on syntactic embedding for low-resource ED of new event types.
- We provide an experimental study on ACE 2005 and MAVEN [35] datasets to demonstrate that our proposed SynED model achieves better performance for new event types in low-resource settings.
- We conduct careful analyses to demonstrate the different roles of syntactic and textual embeddings for low-resource ED.

An overview of this paper is organized as follows. In Section 2, we review the recent related work on event detection and syntactic features. In Section 3, we introduce our proposed low-resource ED method with syntactic embedding called SynED in detail. Section 4 shows the details of the experimental setting and results. In Section 5, we conduct empirical analysis to verify the effectiveness of the method and further discuss the mechanism of the method. Finally, we summarize the conclusion and future work in Section 6.

2. **Related Work.**

2.1. **Event detection methods.**

Most of the previous studies on event detection are fully supervised methods, which can be divided into feature-based methods and neural network-based methods. Feature-based ED methods employ manual-designed features, such as syntactic feature [8], sentiment polarity feature [36], document-level feature [9], entity-level feature [10] and global feature [11]. Neural network-based methods adopt the deep neural architectures which depend on large-scale annotation data, such as convolution neural network (CNN) [1], recurrent neural network (RNN) [12, 15], graph convolution neural network (GCN) [14, 16-18] and hierarchical attention network (HAN) [19].

Other researchers also explored low-resource ED methods. Deng et al. [24], Lai et al. [25], Shen et al. [26] reconstructed the low-resource ED as a few-shot learning task that can be resolved with meta-learning. Wang et al. [23] applied an adversarial training mechanism to iteratively identifying informative instances from a large event-related candidate set. Tong et al. [27] improved the low-resource ED via open-domain trigger knowledge. Liu et al. [28], Du and Cardie [29] enhanced the compatibility of ED models for low-resource event types by casting ED as a machine reading comprehension (MRC) task. Meanwhile, other recent studies [21, 22, 30-33] tended to address the low-resource ED by modeling event ontology with event mentions or event type structure.

In this paper, we propose combining the feature-based approaches and event type ontology approaches, and utilizing the generalization of syntactic features to enhance the ontology representation of new event types. Comparatively to the prior low-resource ED approaches, our proposed SynED method can better handle new event types with little training data by introducing the cross-domain syntactic features.

2.2. **Syntactic features.**

As an essential linguistic element, syntactic features are widely employed in studies to provide more comprehensive semantic information, including speech recognition [37], prosodic event detection [38], text classification [39], named entity recognition [40], author recognition [41], emotion recognition [42], etc. The previous syntactic features are typically created manually for particular applications, making their expansion challenging. In this paper, we utilize the transformer network's potent modeling capabilities to automatically learn syntactic embedding from syntactic tags, which has better task adaptability and expansion ability.

3. **Methodology.** In this paper, we propose a low-resource ED model called SynED with three modules: 1) Syntactic Embedding Construction, 2) Event Trigger Detection, and 3) Event Trigger Classification. The detailed architecture of SynED with running examples is illustrated in Figure 1.

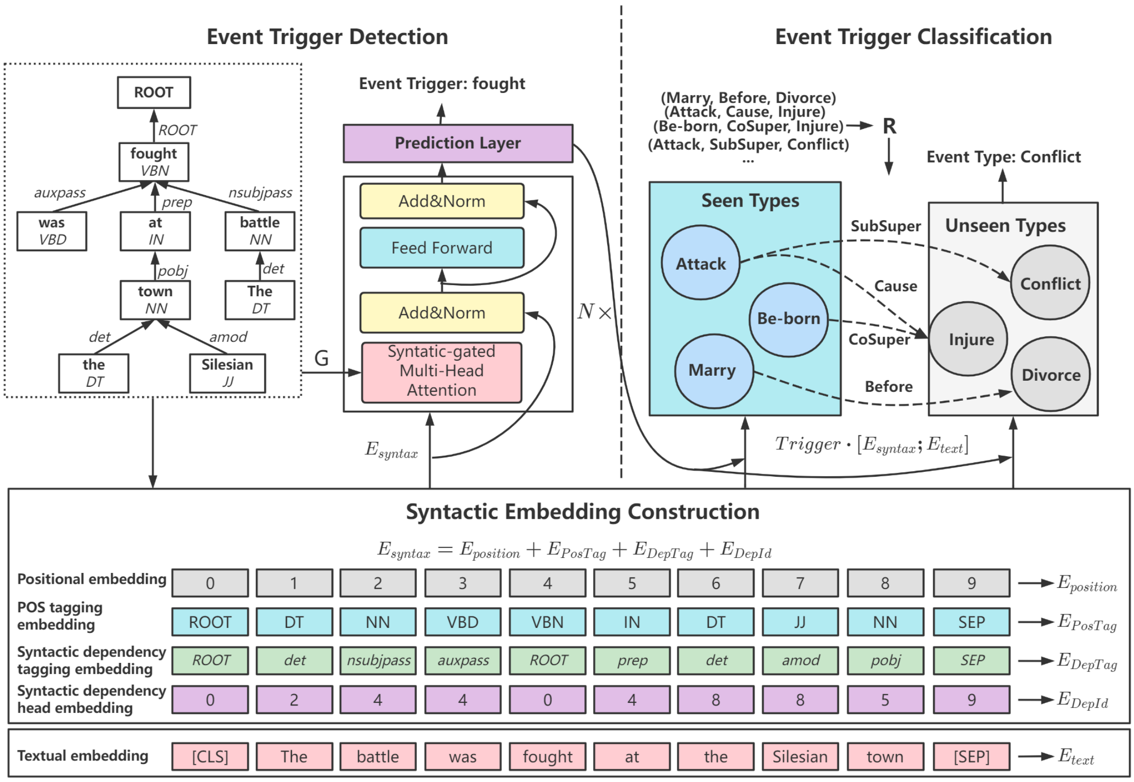


FIGURE 1. Overview of our proposed SynED

3.1. **Syntactic Embedding Construction.** We obtain syntactic information through the SpaCy library and construct corresponding embeddings as follows.

- Positional embedding $E_{position}$: We use the same positional embedding in the pre-trained BERT-base model [43]. Positional embedding helps to model the temporal connection of tokens.
- POS tagging embedding E_{PosTag}: We construct a learnable embedding for each POS tag, e.g., Determiner (DT), and Noun (NN). The POS tag identifies the attribute and purpose of the token.
- Syntactic dependency tagging embedding E_{DepTag}: We construct a learnable embedding for each dependency tag between tokens and their head, e.g., nominal subject ($nsubj$), and prepositional object ($pobj$). The dependency tag determines the role of the token in the sentence.
- Syntactic dependency head embedding E_{DepId}: We construct a learnable embedding for the head id of each token, e.g., the head of "*battle (id: 2)*" is "*fought (id: 4)*". The head id indicates the level of the token in the syntactic dependency tree.

For a given token, we construct its syntactic embedding E_{syntax} by summing the corresponding positional, POS tagging, syntactic dependency tagging, and head embeddings:

$$E_{syntax} = E_{position} + E_{PosTag} + E_{DepTag} + E_{DepId} \tag{1}$$

The above embedding dictionaries are initialized to a normal distribution and updated during event detection training.

3.2. **Event Trigger Detection.** Given the syntactic embedding E_{syntax} of an event mention, we adopt a syntactic-gated Transformer model to get a syntactic-aware representation H. Compared with the original Transformer, the syntactic-gated Transformer model adopts a gated matrix G in the self-attention layer:

$$H = \mathit{SoftMax}\left(\frac{\left(Q \cdot K^T\right) \times G}{\sqrt{d_k}}\right) \times V \tag{2}$$

where $Q = W^Q H$, $K = W^K H$, $V = W^V H$, and W^Q, W^K, W^V are all learnable parameters, and H denotes the hidden state of Transformer with initial state $H^{(0)} = E_{syntax}$. The gated matrix G denotes the connection matrix of the syntactic dependency tree. $\times$ and $\cdot$ denote matrix multiplication and element-wise multiplication, respectively.

Then, the syntactic-aware representation H obtained by the syntactic-gated Transformer is fed to a multi-layer perceptron (MLP) to get the trigger prediction *Trigger*:

$$\mathit{Trigger} = \mathit{SoftMax}(\mathit{MLP}(H)) \tag{3}$$

We adopt cross entropy as the loss function for trigger detection (TD):

$$\mathcal{L}_{TD} = -\sum_{i=0}^{N} y_i^{trigger} \log(\mathit{Trigger}_i) \tag{4}$$

where $y^{trigger}$ denotes the ground-truth label for trigger prediction and N is the length of input sequence.

3.3. **Event Trigger Classification.** We concatenate syntactic and textual representations and fuse them through trigger prediction *Trigger* to obtain trigger representation:

$$E_{\mathit{Trigger}} = \mathit{Trigger} \cdot [E_{syntax}; E_{text}] \tag{5}$$

where the textual representation E_{text} is obtained from the word embeddings of the pre-trained BERT-base [43].

We then initialize the event type prototype P_k for type k with its average instance embedding and event type description embedding.

$$P_k^{(0)} = \frac{1}{1 + |\mathcal{E}_k|}\left(\sum_{e \in \mathcal{E}_k} e + E_{type}^{(k)}\right) \tag{6}$$

where $\mathcal{E}_k$ and $E_{type}^{(k)}$ denote the trigger embeddings of all event mentions and the description embedding for event type k, respectively.

Modeling the internal correlation between different domains has proved to be helpful for knowledge transfer from the source domain to the target domain [22, 44]. Therefore, we update the event type prototype using the correlation of event types, which we refer to as the Correlation Inference mechanism. Here, we use a relational transition matrix $R_r \in \mathbb{R}^{d \times d}$ to model the correlation of event types, which has been proven great robustness in low-resource settings [45].

$$P_k^* = P_k + \lambda \sum_{(k,r,j) \in \mathcal{O}} P_j R_r \tag{7}$$

where a triple $(k, r, j) \in \mathcal{O}$ denotes there is a correlation r between event types k and j, and $\mathcal{O}$ is a set of all triples, and λ is a hyperparameter.

We compute the likelihood of the corresponding event type for event trigger embedding $E_{Trigger}$, denoted by

$$P(type = k) = \frac{\exp\left(-\left\|E_{trigger} - P_k^*\right\|\right)}{\sum_{j=0}^{N_t} \exp\left(-\left\|E_{trigger} - P_j^*\right\|\right)} \tag{8}$$

where $\|\cdot\|$ denotes Euclidean distance, and N_t denotes the number of event types.

We adopt cross entropy as the loss function for Event Trigger Classification (TC):

$$\mathcal{L}_{TC} = -\sum_{i=0}^{N_t} y_k \log(P(type = k)) \tag{9}$$

Besides, we compute the loss function for event type correlation by minimizing the correlation transfer distance and maximizing the inter-class distance:

$$\mathcal{L}_{TR} = -\frac{1}{|\mathcal{O}|} \sum_{(k,r,j)\in\mathcal{O}} \log \frac{P_j R_r P_k}{|P_j R_r| \cdot |P_k|} + \frac{2}{N_t(N_t - 1)} \sum_{k \neq j} \log \frac{P_j P_k}{|P_j| \cdot |P_k|} \tag{10}$$

where $|\cdot|$ denotes Euclidean norm.

The final loss function for Event Trigger Classification is calculated by

$$\hat{\mathcal{L}}_{TC} = \mathcal{L}_{TC} + \alpha \cdot \mathcal{L}_{TR} \tag{11}$$

where α is a hyperparameter.

4. **Experiment.**

4.1. **Data.**

Our experiments are set up on the widely studied ACE 2005 dataset and the newly released large-scale MAVEN [35] dataset. The ACE 2005 dataset contains 33 event types and 5,349 instances, while the MAVEN [35] dataset contains 168 event types and 118,732 instances. To evaluate the performance of our proposed SynED method for new event types, we split the dataset into a training set, validation set and test set with ratios of 0.8, 0.1, and 0.1 respectively, and select 20% of the event types as new event types, ensuring that the training set does not contain data of new event types.

4.2. **Implementation details.**

In SynED, the hidden layer size of the embedding layer and Transformer is set to $d = 768$, and the maximum sequence length is set to $l = 128$. During the training phase, the Adam optimizer is applied, where the learning rate is set to 2×10^{-5} and the batch size is set to 64. The hyperparameters of λ and α are set to 0.1 and 0.5, respectively.

All experiments are arranged on 4× NVIDIA Tesla P100 GPUs. For each model, we trained a total of 30 epochs, which took about 206 minutes. In the test phase, we select the model that performs best in the verification set, and then obtain the evaluation results on the test set. We conducted five repeated trials with different random seeds to calculate the mean and confidence interval of all results.

We evaluate the performance of ED using Prediction (P), Recall (R), and F1 score (F).

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F = \frac{2PR}{P + R} \tag{12}$$

where true positive (TP) denotes the number of positive samples predicted correctly, false positive (FP) denotes the number of negative samples which are predicted as positive, and false negative (FN) denotes the number of positive samples which are predicted as negative.

4.3. **Baselines.** We compare our method with many low-resource ED models:

- **Dynamic-memory-based prototypical network (DMBPN) [24]:** a low-resource ED model based on prototype which reformulates ED as a few-shot learning task and exploits dynamic memory network (DMN) to learn better prototypes for new event types with a little annotation;
- **Enrichment knowledge distillation (EKD) [27]:** a knowledge-enhanced model which leverages external open-domain trigger knowledge to improve the event detection for new event types without annotation;
- **Zero-shot event extraction (ZSEE) [21]:** a zero-shot model which learns the event prototype for new event types through the structure of event type without annotation.
- **Semi-supervised vector quantized variational autoencoder (SS-VQ-VAE) [30]:** a semi-supervised ED framework which automatically learns discrete latent type representations for each type;
- **OntoED [22]:** a low-resource ED model with ontology embedding which enriches the event ontology for new event types with connections between event types.

We have created a number of variants of our proposed SynED model, as shown below, to make it easier to investigate how various modules and features introduced in Section 3 affect the model. The corresponding experimental results are discussed in Section 5.2 and Section 5.3.

- **SynED w/o SynGated:** the SynED model without the syntactic-gated mechanism in the Event Trigger Detection module.
- **SynED w/o Ontology Initialization:** the SynED model without the initialization of event ontology in the Event Trigger Classification module.
- **SynED w/o Correlation Inference:** the SynED model without the Correlation Inference mechanism in the Event Trigger Classification module.
- **SynED w/o Ontology Learning:** the SynED model which treats low-resource ED as a sequence classification task instead of learning event ontology.
- **SynED w/o PosTag:** the SynED model without the POS tagging embedding in the Syntactic Embedding Construction module.
- **SynED w/o DepTag:** the SynED model without the syntactic dependency tagging embedding in the Syntactic Embedding Construction module.
- **SynED w/o DepId:** the SynED model without the syntactic dependency head embedding in the Syntactic Embedding Construction module.
- **SynED$_{text}$:** the SynED model which uses textual embedding as the input of the Event Trigger Detection module instead.
- **SynED$_{random}$:** the SynED model which uses the unordered textual embedding as the input of the Event Trigger Detection module instead.

4.4. **Experimental results.** Table 2 reports the ED evaluation results of the new event types on the ACE 2005 dataset and the MAVEN [35] dataset. We can see that our proposed SynED method achieves better performance compared to baselines, demonstrating its efficacy in handling new event types without annotation data. Our proposed SynED significantly exceeds baseline models on the MAVEN [35] dataset, but shows a weak advantage with large volatility on ACE 2005 dataset. We analyzed that the ACE 2005 dataset, compared with the large-scale MAVEN [35] dataset, has a smaller scale and more incomplete event types, which leads to the invalidity of the knowledge migration method based on event type correlation to the ACE 2005 dataset. Therefore, our following analysis experiments are based on the large-scale MAVEN [35] dataset.

TABLE 2. The ED evaluation results of the new event types on the ACE 2005 dataset and the MAVEN [35] dataset. P (%), R (%) and F (%) stand for precision, recall, and F1-score, respectively.

Model	ACE 2005			MAVEN		
	P	**R**	**F**	**P**	**R**	**F**
DMBPN [24]	12.62±1.69	17.62±1.86	13.51±0.61	24.58±0.55	19.82±0.44	22.10±0.14
EKD [27]	30.38±1.09	30.78±0.99	30.58±0.34	36.28±0.63	41.31±0.43	38.77±0.21
ZSEE [21]	46.72±2.13	34.16±1.06	36.86±1.14	44.28±0.60	40.12±0.53	42.08±0.04
SS-VQ-VAE [30]	43.26±2.36	47.54±2.03	39.78±2.17	44.18±0.30	40.52±0.26	42.22±0.15
OntoED [22]	**50.22±5.40**	36.56±2.18	41.68±2.04	42.36±0.33	46.08±0.52	44.16±0.99
SynED	40.56±2.23	**47.81±0.67**	**43.05±0.54**	**69.73±0.20**	**65.78±0.84**	**68.06±0.44**

5. **Discussion.** We have created more analysis experiments and covered them in-depth in this section in order to get a thorough assessment of our SynED method and comprehend how syntactic embedding works.

5.1. **Impact of the number of training data.** We evaluate the performance of several low-resource ED methods for new event types with different ratios of training data, as shown in Figure 2. Experimental results demonstrate the superiority of our proposed SynED method in low-resource settings compared to previous ED methods. We can see that the SynED model achieves competitive performance with fully supervised settings, using only a very small amount of training data (1%). In contrast, other baselines require at least 30% of the training data, which demonstrates the benefits of the SynED method using syntactic embedding in overcoming data reliance compared with prior studies.

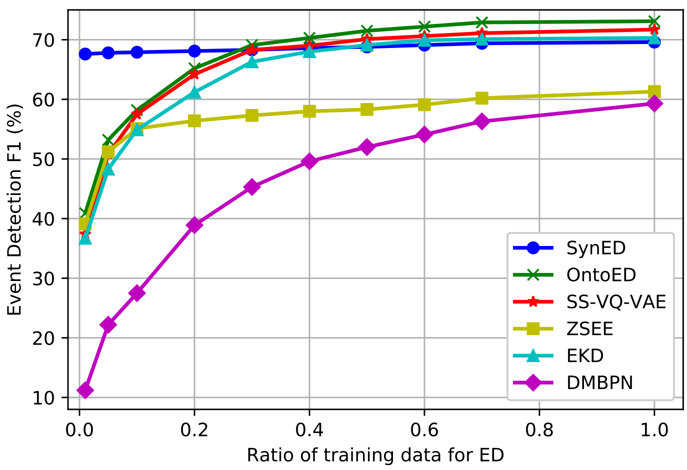


FIGURE 2. Results on different ratios of ED training data for new event types

5.2. **Ablation study: Impact of each module.** To assess the effect of each module, we remove some mechanisms (e.g., the syntactic-gated mechanism in the Event Trigger Detection module, the initialization of event ontology, the Correlation Inference mechanism, and the event ontology learning in the Event Trigger Classification module) applied in SynED method and evaluate F1 score for new event types in zero-shot and fully supervised settings as shown in Table 3. We observe that the removal of either of those mechanisms

TABLE 3. F1 (%) score of ablation study of each module

Model	Zero-shot	Fully supervised
SynED	68.06±0.44	69.42±0.13
w/o SynGated	67.52±0.19	**67.98±0.11**
w/o Ontology Initialization	38.74±0.41	69.15±0.23
w/o Correlation Inference	29.09±0.54	69.41±0.16
w/o Ontology Learning	**0.00±0.00**	69.92±0.59

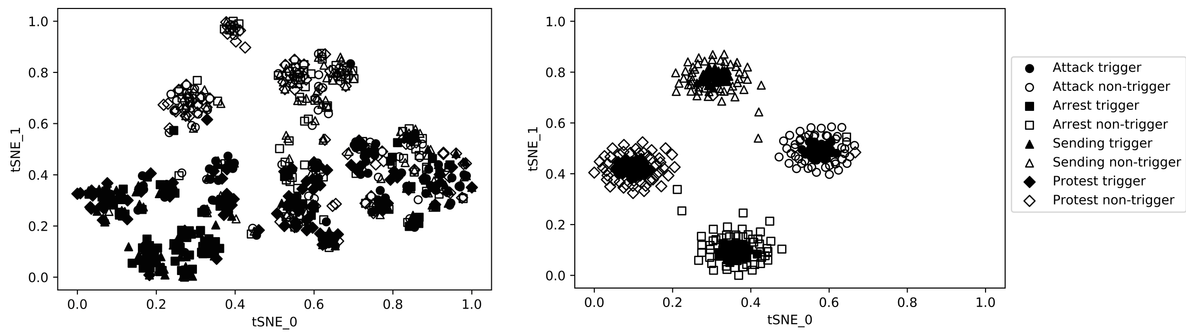
TABLE 4. F1 (%) score of ablation study of syntactic embedding

Model	Zero-shot	Fully supervised
SynED	68.06±0.44	69.42±0.13
w/o PosTag	65.52±0.27	66.72±0.29
w/o DepTag	65.78±0.45	67.86±0.13
w/o DepId	67.38±0.19	68.28±0.22
SynED$_{text}$	44.94±0.11	72.48±0.31
SynED$_{random}$	**37.96±1.24**	**66.42±0.61**

causes a significant performance reduction (e.g., 0.54% ↓ v.s. 29.32% ↓ v.s. 38.97% ↓ and 68.06% ↓) in zero-shot settings, especially SynED w/o Ontology Learning. We guess the phenomenon is due to the overfitting of the sequence classification model to seen event types. In the fully supervised settings, the removal of syntactic-gated mechanism, ontology initialization or correlation inference results in variable degrees of performance degradation (e.g., 1.44% ↓ v.s. 0.27% ↓ and 0.01% ↓), but the removal of ontology learning results in performance improvement (e.g., 0.50% ↑). We analyze that the training data of new event types in the fully supervised settings is very sufficient, which minimizes the effects of the initialization and correlation reasoning of event ontology, and improves the performance of the sequence classification model.

5.3. **Ablation study: Impact of syntactic embedding.** To assess the impact of syntactic embedding on ED, we evaluate the ED performance using different embeddings as input, as shown in Table 4. We discover that significant performance reduction occurs when POS tagging, dependency tagging, or dependency heads are removed (e.g., 2.54% ↓ v.s. 2.28% ↓ and 0.68% ↓). Compared with the SynED with syntactic embedding, the SynED with textual embedding (SynED$_{text}$) shows low accuracy (e.g., 23.12% ↓) for new event types in zero-shot settings but better performance (e.g., 3.06% ↑) in the fully supervised settings. Experimental results demonstrate that syntactic information is general for different event types, which makes it easier to transfer knowledge between event types with the loss of some type-specific knowledge. We also evaluate the ED performance of the SynED with unordered textual embedding (SynED$_{random}$) which can be regarded as the complete removal of syntactic information. The SynED$_{text}$ model outperforms the SynED$_{random}$ model in both zero-shot and fully supervised settings (e.g., 6.98% ↑ and 6.06% ↑), highlighting the importance of syntactic information for ED.

5.4. **Visualization study: How does syntactic embedding work?** To figure out the mechanism of syntactic embedding in new event type detection, we randomly sampled some samples from four event types and visualized their syntactic and textual embeddings as shown in Figure 3 through T-SNE [46]. We observed that syntactic embeddings distinguish between triggers and non-triggers but mingle between different event types,



(a) Syntactic embedding (b) Textual embedding

FIGURE 3. The embedding distribution of syntactic embedding and textual embedding of four event types (e.g., Attack, Arrest, Sending and Protest)

while textual embeddings present the opposite. Therefore, in the Event Trigger Detection module, only using syntactic embedding can well distinguish triggers from non-triggers without additional annotation, while in the Event Trigger Classification module, adding textual embedding makes different event types distinguished easily.

6. **Conclusion.** In this paper, we propose a novel syntactic-based low-resource ED model called *SynED*. We first propose a Syntactic Embedding Construction module to improve the detection of new event types through the generalization of syntactic information. For the Event Trigger Detection module, we propose a syntactic-gated Transformer as the trigger extractor. For the Event Trigger Classification module, we propose to learn the ontology of new event types through knowledge transfer based on type correlation. Experimental results on ACE 2005 and MAVEN [35] datasets demonstrate the effectiveness of our proposed SynED method for low-resource ED of new event types. We also explore the mechanism of syntactic and textual embeddings in new event type detection and reveal the important role of introducing syntactic information for low-resource ED.

In this paper, we introduce syntactic information by constructing learnable syntactic embedding and applying a syntactic-gated Transformer model, which is proved to be effective for detecting new event types under low resource settings. In future work, we intend to investigate more effective ways to utilize syntactic information. For example, we will try to use a graph neural network to learn the graph structure representation of the syntactic dependency tree.

Acknowledgment. This work is partially supported by The Youth Innovation Promotion Association of the Chinese Academy of Sciences (E1291902), Jun Zhou (2021025).

REFERENCES

[1] Y. Chen, L. Xu, K. Liu, D. Zeng and J. Zhao, Event extraction via dynamic multi-pooling convolutional neural networks, *Proc. of the Annual Meeting of the Association for Computational Linguistics*, pp.167-176, 2015.

[2] D. Li, L. Huang, H. Ji and J. Han, Biomedical event extraction based on knowledge-driven Tree-LSTM, *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp.1421-1430, 2019.

[3] X. D. Wang, L. Weber and U. Leser, Biomedical event extraction as multi-turn question answering, *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pp.88-96, 2020.

[4] S. Deng, N. Zhang, W. Zhang, J. Chen, J. Z. Pan and H. Chen, Knowledge-driven stock trend prediction and explanation via temporal convolutional network, *Companion Proceedings of the 2019 World Wide Web Conference*, pp.678-685, 2019.

[5] X. Liang, D. Cheng, F. Yang, Y. Luo, W. Qian and A. Zhou, F-HMTC: Detecting financial events for investment decisions based on neural hierarchical multi-label text classification, *The 29th International Joint Conference on Artificial Intelligence and 17th Pacific Rim International Conference on Artificial Intelligence*, pp.4490-4496, 2020.

[6] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su and J. Gao, EANN: Event adversarial neural networks for multi-modal fake news detection, *The 24th ACM SIGKDD International Conference*, pp.849-857, 2018.

[7] M. N. Nikiforos, S. Vergis, A. Stylidou, N. Augoustis, K. L. Kermanidis and M. Maragoudakis, Fake news detection regarding the Hong Kong events from tweets, in *Artificial Intelligence Applications and Innovations. AIAI 2020 IFIP WG 12.5 International Workshops. AIAI 2020. IFIP Advances in Information and Communication Technology*, I. Maglogiannis, L. Iliadis and E. Pimenidis (eds.), Cham, Springer, 2020.

[8] D. Ahn, The stages of event extraction, *Proc. of the Workshop on Annotating and Reasoning about Time and Events*, Sydney, Australia, pp.1-8, 2006.

[9] H. Ji and R. Grishman, Refining event extraction through cross-document inference, *Proc. of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp.254-262, 2008.

[10] Y. Hong, J. Zhang, B. Ma, J. Yao, G. Zhou and Q. Zhu, Using cross-entity inference to improve event extraction, *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, OR, USA, pp.1127-1136, 2011.

[11] Q. Li, H. Ji and L. Huang, Joint event extraction via structured prediction with global features, *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics*, pp.73-82, 2013.

[12] T. H. Nguyen, K. Cho and R. Grishman, Joint event extraction via recurrent neural networks, *Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp.300-309, 2016.

[13] B. Yang and T. Mitchell, Joint extraction of events and entities within a document context, *Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp.289-299, 2016.

[14] X. Liu, Z. Luo and H. Huang, Jointly multiple events extraction via attention based graph information aggregation, *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pp.1247-1256, 2018.

[15] L. Sha, F. Qian, B. Chang and Z. Sui, Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction, *AAAI Conference on Artificial Intelligence*, 2018.

[16] J. Liu, Y. Chen, K. Liu and J. Zhao, Neural cross-lingual event detection with minimal parallel resources, *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Hong Kong, China, pp.738-748, 2019.

[17] H. Yan, X. Jin, X. Meng, J. Guo and X. Cheng, Event detection with multi-order graph convolution and aggregated attention, *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp.5765-5769, 2019.

[18] S. Cui, B. Yu, T. Liu, Z. Zhang, X. Wang and J. Shi, Edge enhanced graph convolution networks for event detection with syntactic relation, *Proc. of the Conference on Empirical Methods in Natural Language Processing (Findings)*, pp.2329-2339, 2020.

[19] S. Shen, G. Qi, Z. Li, S. Bi and L. Wang, Hierarchical Chinese legal event extraction via pedal attention mechanism, *Proc. of the 28th International Conference on Computational Linguistics*, pp.100-113, 2020.

[20] D. Lou, Z. Liao, S. Deng, N. Zhang and H. Chen, MLBiNet: A cross-sentence collective event detection network, *Proc. of the Annual Meeting of the Association for Computational Linguistics*, 2021.

[21] L. Huang, H. Ji, K. Cho, I. Dagan, S. Riedel and C. R. Voss, Zero-shot transfer learning for event extraction, *Proc. of the Annual Meeting of the Association for Computational Linguistics*, pp.2160-2170, 2018.

[22] S. Deng, N. Zhang, L. Li, H. Chen, H. Tou, M. Chen, F. Huang and H. Chen, OntoED: Low-resource event detection with ontology embedding, *Proc. of the Annual Meeting of the Association for Computational Linguistics*, 2021.

[23] X. Wang, X. Han, Z. Liu, M. Sun and P. Li, Adversarial training for weakly supervised event detection, *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp.998-1008, 2019.

[24] S. Deng, N. Zhang, J. Kang, Y. Zhang, W. Zhang and H. Chen, Meta-learning with dynamic-memory-based prototypical network for few-shot event detection, *Proc. of the 13th International Conference on Web Search and Data Mining*, Houston, TX, USA, pp.151-159, 2020.

[25] V. D. Lai, T. H. Nguyen and F. Dernoncourt, Extensively matching for few-shot learning event detection, *Proc. of the Annual Meeting of the Association for Computational Linguistics*, pp.38-45, 2020.

[26] S. Shen, T. Wu, G. Qi, Y.-F. Li, G. Haffari and S. Bi, Adaptive knowledge-enhanced Bayesian meta-learning for few-shot event detection, *Findings of the Association for Computational Linguistics*, 2021.

[27] M. Tong, B. Xu, S. Wang, Y. Cao, L. Hou, J. Li and J. Xie, Improving event detection via open-domain trigger knowledge, *Proc. of the Annual Meeting of the Association for Computational Linguistics*, pp.5887-5897, 2020.

[28] J. Liu, Y. Chen, K. Liu, W. Bi and X. Liu, Event extraction as machine reading comprehension, *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pp.1641-1651, 2020.

[29] X. Du and C. Cardie, Event extraction by answering (almost) natural questions, *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pp.671-683, 2020.

[30] L. Huang and H. Ji, Semi-supervised new event type induction and event detection, *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pp.718-724, 2020.

[31] R. Lima, B. Espinasse and F. Freitas, OntoILPER: An ontology- and inductive logic programming-based system to extract entities and relations from text, *Knowl. Inf. Syst.*, vol.56, no.1, pp.223-255, 2018.

[32] R. Lima, B. Espinasse and F. Freitas, A logic-based relational learning approach to relation extraction: The ontoilper system, *Eng. Appl. Artif. Intell.*, vol.78, pp.142-157, 2019.

[33] M. Li, Q. Zeng, Y. Lin, K. Cho, H. Ji, J. May, N. Chambers and C. R. Voss, Connecting the dots: Event graph schema induction with path language modeling, *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pp.684-695, 2020.

[34] M. Sims, J. H. Park and D. Bamman, Literary event detection, *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp.3623-3634, 2019.

[35] X. Wang, Z. Wang, X. Han, W. Jiang, R. Han, Z. Liu, J. Li, P. Li, Y. Lin and J. Zhou, MAVEN: A massive generic domain event detection dataset, *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pp.1652-1671, 2020.

[36] K. V. Rajan, V. R. Uthariaraj and K. Paul, Sentiment pivot approach for event detection from Twitter stream, *ICIC Express Letters, Part B: Applications*, vol.12, no.9, pp.789-796, DOI: 10.24507/icicelb.12.09.789, 2021.

[37] H.-K. J. Kuo, L. Mangu, A. Emami, I. Zitouni and Y.-S. Lee, Syntactic features for Arabic speech recognition, *2009 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp.327-332, DOI: 10.1109/ASRU.2009.5373470, 2009.

[38] J. H. Jeon and Y. Liu, Automatic prosodic events detection using syllable-based acoustic and syntactic features, *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp.4565-4568, DOI: 10.1109/ICASSP.2009.4960646, 2009.

[39] I. Fahmi and G. Bouma, Learning to identify definitions using syntactic features, *Proc. of the Workshop on Learning Structured Information in Natural Language Applications*, 2009.

[40] H. Pham and L.-H. Phuong, The importance of automatic syntactic features in Vietnamese named entity recognition, *Proc. of the 31st Pacific Asia Conference on Language, Information and Computation*, 2017.

[41] J. Soler and L. Wanner, On the relevance of syntactic and discourse features for author profiling and identification, *Proc. of the 15th Conference of the European Chapter of the Association for Computational*, 2017.

[42] A. S. Hosseini, Sentence-level emotion mining based on combination of adaptive meta-level features and sentence syntactic features, *Engineering Applications of Artificial Intelligence*, vol.65, pp.361-374, 2017.

[43] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp.4171-4186, 2019.

[44] X. Zhang, P. Zhang and Y. Yan, Multi-step dialogue state tracker with schema-guided graph convolutional network, *International Journal of Innovative Computing, Information and Control*, vol.17, no.6, pp.1955-1971, DOI: 10.24507/ijicic.17.06.1955, 2021.
[45] W. Zhang, B. Paudel, L. Wang, J. Chen, H. Zhu, W. Zhang, A. Bernstein and H. Chen, Iteratively learning embeddings and rules for knowledge graph reasoning, *The World Wide Web Conference*, pp.2366-2377, 2019.
[46] L. van der Maaten and G. Hinton, Visualizing data using t-SNE, *Journal of Machine Learning Research*, pp.2579-2605, 2008.

Author Biography

Ruiliu Fu received the B.E. degree in Electronic Engineering from Tsinghua University, China, 2017. He is currently a Ph.D. student in Institute of Acoustics, Chinese Academy of Sciences, China. His research interests include question answering system, interpretability in natural language understanding, event detection, and dialogue system.

Han Wang received the B.E. degree in Communication Engineering from Minzu University of China in 2016. He is currently a Ph.D. student at the Institute of Acoustics, Chinese Academy of Sciences and University of Chinese Academy of Sciences, Beijing, China. His research interests include incremental/lifelong/continual learning, dialogue system, and text classification.

Xuejun Zhang received the B.E. degree in Electronic Communication Engineering from North China Institute of Science and Technology, Hebei, China, in 2012 and the M.S. degree in Communication Engineering from Beijing Institute of Technology, Beijing, China, in 2015. She is currently pursuing the Ph.D. degree in Natural Language Processing from Institute of Acoustics, Chinese Academy of Sciences (CAS), Beijing, China. From 2015 to 2018, she was a Research Assistant with the Institute of Acoustics, Chinese Academy of Sciences (CAS), Beijing, China. Her current research interest lies in spoken language understanding, dialog state tracking, and dialogue decision.

Jun Zhou received the Ph.D. degree in Information Engineering from Institute of Information Engineering, Chinese Academy of Sciences, China, 2017. He is currently an associate researcher at Institute of Acoustics, Chinese Academy of Sciences, China. His research interests include big data computing, dialogue system, and question answering system.

Yonghong Yan received the B.E. degree in Electronic Engineering from Tsinghua University, China, 1990; the Ph.D. degree in Computer Science and Engineering from Oregon Graduate Institute of Science and Technology (OGI), USA, 1995.

Prof. Yan is currently a full-time professor at Institute of Acoustics, Chinese Academy of Sciences, China. His main research interests include speech signal processing, speech recognition, spoken language system and multi-mode system, man-machine interface technology. He has published over 200 papers on well-known journals. He is currently the executive director of China Acoustic Society, the deputy editor in chief of Applied Acoustics, the editorial board of acoustic magazine and the Journal of Computer Science and Technology.