Lexical and Compositional Stream Learning for Event Detection with Sememe Knowledge

Jiale Yuan, Xin Xin and Ping Guo

May 12, 2020

Lexical and Compositional Stream Learning for Event Detection with Sememe Knowledge

1st Jiale Yuan
School of Computer Science and Technology
Beijing Institute of Technology
Beijing, China
3120191071@bit.edu.cn

2nd Xin Xin*
School of Computer Science and Technology
Beijing Institute of Technology
Beijing, China
xxin@bit.edu.cn

3rd Ping Guo
School of Systems Science
Beijing Normal University
Beijing, China
pguo@ieee.org

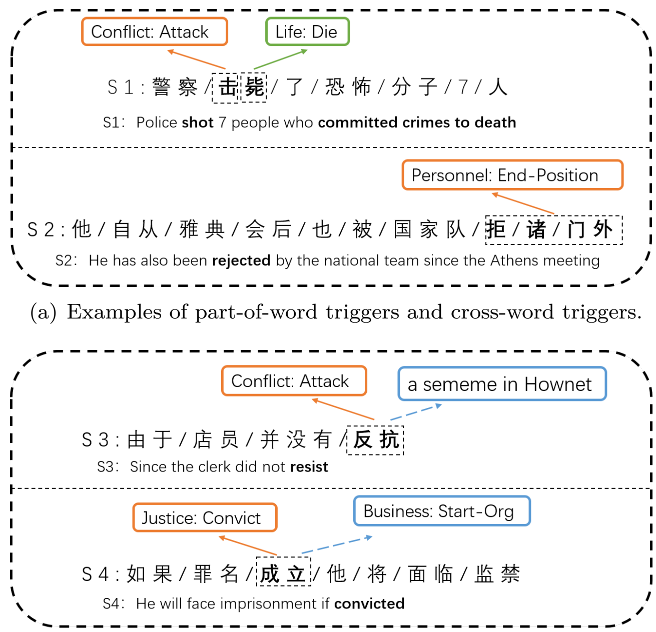
***Abstract*—As one of the most significant subtasks for event extraction, event detection(ED) aims to identify the trigger words in a sentence and classify them with correct event types. Most methods in previous work rely on various neural networks to extract trigger features automatically which still suffer a lot from word-trigger mismatch and disability of sparse triggers detecting, especially in Chinese corpus. In this paper, we propose a lexical and compositional stream learning approach to alleviate these two limitations in ED task with sememes in HowNet as the external knowledge base. Concretely, we employ convolutional neural network (CNN) to learn lexical representation and compositional representation separately, and we consolidate event sememe information into structural features where the event sememe embeddings provide sememe trigger clues in sentence-level and word-sememe-type tertiary structure enriches the compositional features. Then we fuse both of them into a hybrid representation to achieve trigger identification and event type classification. Experiments conducted on ACE2005 dataset show our model outperforms the state-of-the-art method especially for event type classifier.**

***Index Terms*—event detection, event sememe, HowNet, CNN**

I. Introduction

Event detection (ED) is the fundamental procedure of event extraction, which is designed to locate the trigger words in a sentence and classify the trigger words with related event types. Doing research of ED could make a difference to automatic information extraction and text understanding [1]. Generally, the experiment of ED task is partitioned to event trigger identification stage and event type classification stage. The former part aims to dig out trigger words and the latter one aims to specify event types. For example, given a sentence "*Dozens of civilians were killed in this barrage.*", the identifier of model detects "*killed*" as the trigger word and the classifier gives the event type of trigger as "*Conflict:Attack*".

Currently, plenty of methods based on neural network have achieved significant advance in ED task [2]–[5]. However, these works still have limitations in word-trigger mismatch and sparsely labeled trigger detecting. Due to the natural word delimiters missing in Chinese, such two limitations are more apparent because of the imperfect approach of language segmentation.



(a) Examples of part-of-word triggers and cross-word triggers.

(b) Examples of OOV and OOL trigger words.

Fig. 1: Illustrative sentences about event trigger detecting problems. Word boundaries are denoted by slashes. Black bold font indicates trigger words.

We illustrate word-trigger mismatch limitation in Fig. 1(a), the data preprocessing tool sets the word "击毙" (kill and die) in S1 as a token, where "击" (kill) and "毙" (die) trigger different event types. S2 shows a cross-word trigger which contains three tokens while triggering one event type "*Personnel:End-Position*". The two instances in Fig. 1(b) show sparsely labeled trigger words, where "反抗" (resist) in S3 is a trigger occurred in testing corpus but not in training corpus, and the label "*Justice:Convict*" of "成立" (convicted) in S4 never occurs in training corpus. Table I shows the proportion of

*The corresponding author.

TABLE I: Statistics of Different Match Types between (Sememe)Words and Triggers on ACE2005

Datasets	Match type			Sememe Triggers
	Mismatch	*OOV*	*OOL*	
ACE2005	14.61%	8.06%	1.08%	26.11%

mismatch and sparse trigger words on ACE2005 datasets, from the statistical data we can observe that the above limitations make a huge challenge in this task. Following the most previous works, it is not sufficient to transform trigger words into character-wise or word-wise representation using neural network. Some proposed method like [6], [7] dealing with issues in that way could only detect discriminate triggers and miss some ambiguous words and sparsely labeled words. To alleviate the above issues [8] designed a fix-scale span to predict trigger candidates, from another perspective, this approach makes the trigger overlap problem worse.

In this paper, we propose a lexical and compositional stream learning approach, which utilizes sememes in Hownet as an external knowledge. [9], [10] proposed a word representation learning approach using sememe embeddings to obtain appropriate senses for words, but we consider that it is prejudiced for word representation learning because each word sense in different context could be directed by different sememes. To extract trigger information efficiently, we integrate a sememe-trigger structure with annotated event types. Concretely, we select and annotate 443 sememes with possible event types on the basis of the event type annotation rules of ACE2005 and the word-sememe annotation in HowNet, in order to maximize the role which sememes played of in the mission. As shown in table 1, there are 26.11% trigger words in dataset being directly related with sememes, i.e., trigger word "反抗" (resist) in S3 would be missed by traditional method due to its OOV attribute, however , we add association between sememes and words and manually annotate event type "*Conflict:Attack*" for this word. After the sememe consolidation, our model would trigger this word correctly and classify it with right event type.

Further, we decouple a trigger word representation into two streams $R = \mathbf{r}_l \oplus \mathbf{r}_c$ where $\mathbf{r}_l$ is lexical-trigger representation which learned by lexical stream learning approach to capture semantic clues from context, $\mathbf{r}_c$ means the compositional-trigger representation which generated by compositional stream learning approach to capture morphological structure clues of trigger candidates, and $\oplus$ is the fusion gate of model we would explain in the later section. We adopt jointly representation learning with sememes as initial embeddings to generate original word representation for the upper network layer. The lexical stream learning approach detects lexical clues from context, i.e., trigger "成立"(convicted) in S4 would be classified as type "*Justice:Convict*" rather than "*Business:Start-Org*" owing to {罪名 (accusation),..., 监禁 (imprisonment)}. The compositional stream learning approach captures structural clues with specific patterns like "留任", "补任" as pattern "verb+ 任" which trigger the same type "*Personnel:start-position*". Then we add sememe trigger information to word-level representation, strengthening the model's ability to extract trigger clues. The architecture of our model is shown in Fig. 2.

We conduct experiments on ACE2005 dataset to examine the efficiency of our model. Experimental results show that our lexical and compositional stream learning approach outperforms on mismatch problems and sparse triggers with event sememe information.

The main contributions of our paper are as follows:

- We propose a lexical and compositional stream learning approach to capture semantic and structural clues for trigger words in a sentence. Utilizing these two level clues can make a progress in Chinese information extraction tasks.
- We consolidate 306 sememe triggers with potential event type values to capture trigger word features. With the support of word-sememe-type hierarchy structure, our model can dig out the correct trigger tokens matched with apt event type, and alleviate the mismath and sparsely labeled problems. Experimental results show that our model outperforms in ACE2005 dataset.

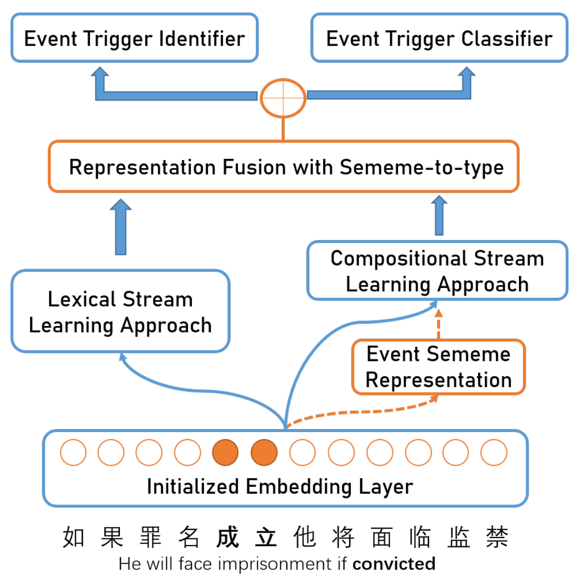


Fig. 2: The architecture of lexical and compositional stream learning approach with sememe knowledge.

II. Related Work

Event Detection is one of the subtask of event extraction, Besides ED, arguments identification and classification is also an important issue in event extraction. An event is defined in a sentence according to its information about what happened, where and to whom, and the temporal message [1]. It suggests the difference between events and common vocabulary. Information extraction scopes would be carried forward rely on ED. Recently, with the widespread application of neural network in NLP mission, features extracted automatically put up a great advantage against the traditional approaches of information extraction. some methods like [2], [4], [11] employ classical neural network flexibly to learn representation from diverse token level in favor of the optimization objective for their model. Concretely, Reference [2] designed a DMCNN framework with a dynamic multi-pooling layer which could capture sentence-level clues. In [12], [13] they brought some new ideas on RNN to extract latent features for task. Familiar with the target concern, Reference [11] focused on feature extraction with a context attention mechanism. The main drawback of these forms is that network mechanism show an implicit feature inflection which is not easy to locate specific information for input sequences. In the meanwhile, the accuracy improvement of event detection is limited owing to finite corpus. Besides, different from English corpus, Chinese event detection suffers a lot in language delimiters. And that is why we dealing with Chinese corpus in task from character-level and word-level like the works in [8], [12].

It is worth mentioning that the development of pretrained language model [14], [15] is beneficial to model initialized embedding learning. Reference [16] proposed an event extraction framework which is based on pretrained language models to generate labeled samples through argument replacement and adjunct tokens rewriting. However, these kind of approaches still miss accuracy in trigger-word mismatch problems and sparsely labeled triggers.

Hownet Knowledge Usage. HowNet knowledge base annotates each concept in Chinese with one or more relevant sememe words. The introduction of HowNet given by [17], [18] makes a overall guidance about the structure of sememe words with senses, and the knowledge database make-up language which used in HowNet. Reference [19] introducts morphological structures with the sememes of Chinese words to infer unknown triggers. The work proposed in [10] aims to improve the semantics of word embeddings by incorporating word sememes into word representation learning. The main idea of the SAT model is that learning original word embeddings for context words, but sememe embeddings for target words via sememe attention which conducts more reliable and accurate word representations. Embedding computing with sememe knowledge shows advantages in NLP downstream tasks. It also shows potential in application of sememes. Reference [20] utilizes sememe knowledge to model semantic compositionality and achieves great performance on intrinsic and extrinsic evaluations. In [21] HowNet is employed as external linguistic knowledge to model multiple senses of polysemous words which alleviate polysemy ambiguity in Chinese relation extraction task. Method put forward in [22] makes use of sememes from lexical semantic resources via hierarchical attention mechanism for aspect extraction. with the help of sememes their unsupervised neural models could explore latent semantic information behind implicit and various expressions. Reference [23] utilizes sememe information in multi-channel reverse dictionary task which predict characteristics of target words from given input queries.

Sememe application in different missions has to consider various adaptation. Unlike sememe information fusion in above methods, we select part of sememes in this paper in order to detect event triggers and event types Specifically. As the features incorporate into sentence-level clues adaptively, the challenge of word-trigger mismatch and unseen trigger types could be alleviated in a soft way for ED task.

III. Methodology

Given a sentence, the model ought to distinguish whether there is a event trigger word in this sentence. If so, the event type of word should be given by event type classifier. In this section, we generalize the limitation of Chinese trigger words detection on Automatic Context Extraction(ACE)2005 as follows:

- One trigger V.S. part-of-word span, which the token in a sentence generated by NLP segmentation toolkits contains not only one event trigger.
- One trigger V.S. cross-of-word span, which the trigger is composed of not only one token in the sentence.
- OOV and OOL. We conduct experiments on ACE2005 dataset, where the different types of trigger words in training and testing sets play an important role in the performance of event detecting model. OOV words is out-of-vocabulary words, which could be detected in testing corpus not in training corpus. OOL words are the out-of-label words, for example, an instance whose (word, type) never occurs in the training corpus but in testing corpus, at the same time it not belongs to OOV.

The former two issues are collectively gathered as word-trigger mismatch problems, and the last one is reflected in sparsely labeled trigger words. The details of lexical and compositional stream learning approach to alleviate such issues are described as following.

As the learning approach introduction, we develop two different aspects to capture both lexical and compositional clues in a sentence. it shows advantages to transform the event trigger words into character-based and word-based hybrid representation where the characters locate the internal compositional structure of event triggers and words describe the semantic context between characters [8], [24].

In this paper, we extract semantic features as r_l and structural features as r_c by convolutional neural network (CNN) model [25], [26] in two substantive part. In order to make exhaustive use of the imformation contained by r_l

and r_c in trigger identifier and event classifier, we design a fuse gate to consolidate the above two representation in task. Due to the event description with sememe in Hownet would be more definite in word-level rather than char-level, so we fuse event sememe clues and sememe-to-type information into word-based representation. As shown in Fig. 2, our model primarily includes the following four stages: token initialized layer to locate apposite embeddings of char-level, word-level and event sememes in a sentence. Lexical and compositional representation learning, which extracts semantic and structural features from different aspects. hybrid representantion learning which employs a gate mechanism to fuse both representations and event sememes memory. And model training about the propability for a instance being trigger candidate. The details of each part would be shown in this section.

A. Token Initialized Layer

HowNet knowledge base [17] annotates a finite sememe set to generalize the concept of words which express different senses in different contexts. Given a sequence $S = \{x_0, x_1, ..., x_n\}$ where $x_i = \{c_i, w_i, s_i\}$ contains three level embeddings regard as characters, words and event sememes, we embed each token x_i except event sememes as $\mathbf{x}_i = [\mathbf{e}_w; \mathbf{e}_p]$ where $\mathbf{e}_w$ is the word embedding of token and $\mathbf{e}_p$ is the relative position embedding to the target token. Following the way of SAT model [10] improving word representation with sememe attention, each token embedding could be generated accurately with sememe-based attention scores.

The function of event sememes which are selected from HowNet is made up of two parts: the first one is to judge whether the concerning target word x_c directs to one or more event sememes, if so, we add event sememe embeddings like:

$$\mathbf{x}^c_{s_i} = e(sem(s^c_i)) \tag{1}$$

where s^c_i represents the ith event sememe embedding of x_c, $\mathbf{x}^c_{s_i}$ is the ith sememe embedding for x_c.

The second part indicates a relationship between event sememes and event types depending on manually labeling information. For example, tuple instance (w, S, T) establishes a word-sememe-type structure where S is event sememe set of word w which we collect from HowNet, T is the possible event type set of sememe trigger, which we manually annotate following the concept of sememes and ACE2005 annotation rules.

The explanation of word-sememe-type tertiary structure is shown in Fig. 3. Concretely, there are 34 labels in event type, we add $tag(i) \in T$ for event sememe only if the sense of sememe word account for this event type. The tag information of sememe word is compressed into fixed length representation as t^c for center word.

B. Lexical and Compositional Representation

We employ convolutional neural network(CNN) to train both lexical stream and compositional strean which is

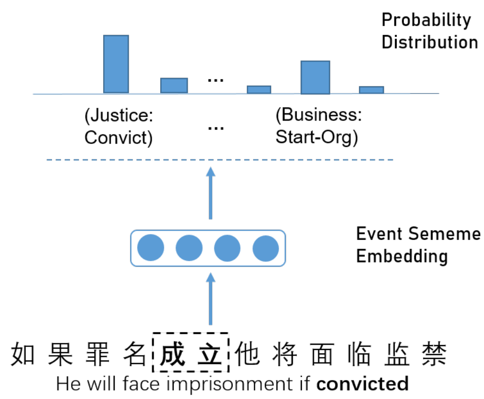


Fig. 3: The demonstration of word-sememe-type tertiary. Second stage is event sememe representation of target word and third stage describes the probability distribution of event type for target word.

similar with the previous work [2], [27]. Before entering the model, we conduct raw context with a fixed window size by curtail redundant sentences and padding for the short ones with predifined tokens.

Lexical Stream Learning. As a sequence like S where $x_i = c_i$ describes single character being tokens, we set filter $\mathbf{W} \in \mathbb{R}^{h \times d}$ for convolutional layer computing a new feature. In detail, considering $\mathbf{x}_i$ is the concatenation of word embeddings and position embeddings of characters and concerning the center word x_c, the local feature captured for character is as follows which is similar with [2]:

$$\mathbf{l}_{ki} = \sigma(\mathbf{w}_k \cdot \mathbf{x}_{i:i+h-1} + \mathbf{b}_k) \tag{2}$$

where $\mathbf{x}_{i:i+h-1}$ refers to the concatenated embeddings of words from x_i to x_{h-1} and h represents the window size of convolutional layer, $\mathbf{w}_k$ indicates the kth filter of $\mathbf{W}$ in convolutional computing, $\mathbf{b}_k \in R$ is a bias term and σ is a non-linear function which we use ReLU here. After the convolutional computing, we employ dynamic multi-pooling method to obtain the max value from different parts in the sentence which is similar with [8]:

$$\begin{aligned} \mathbf{l}^{left}_k &= \max_{i<c} \mathbf{l}_{ki} \\ \mathbf{l}^{right}_k &= \max_{i \geq c} \mathbf{l}_{ki} \end{aligned} \tag{3}$$

In this way the most valuable information could be reserved without missing the max value. Then we concatenate both context features as $\mathbf{l}_{char} = [\mathbf{l}^{left}_k, \mathbf{l}^{right}_k]$. After convolutional feature obtained, we generate lexical representation using lexical gate in char-level with the embeddings $\mathbf{f}_L = \mathbf{l}_{char} \oplus \mathbf{l}_{lex}$, where $\mathbf{l}_{lex}$ is the context embeddings extracted aside by center token:

$$\mathbf{r}_l = \sigma(W_L \mathbf{f}_L + b_L) \tag{4}$$

where $\mathbf{W}_L \in \mathbb{R}^{n \times d_l}$ is the weight matrix, n is the length of feature maps and d_l is the dimension of leixcal features, σ

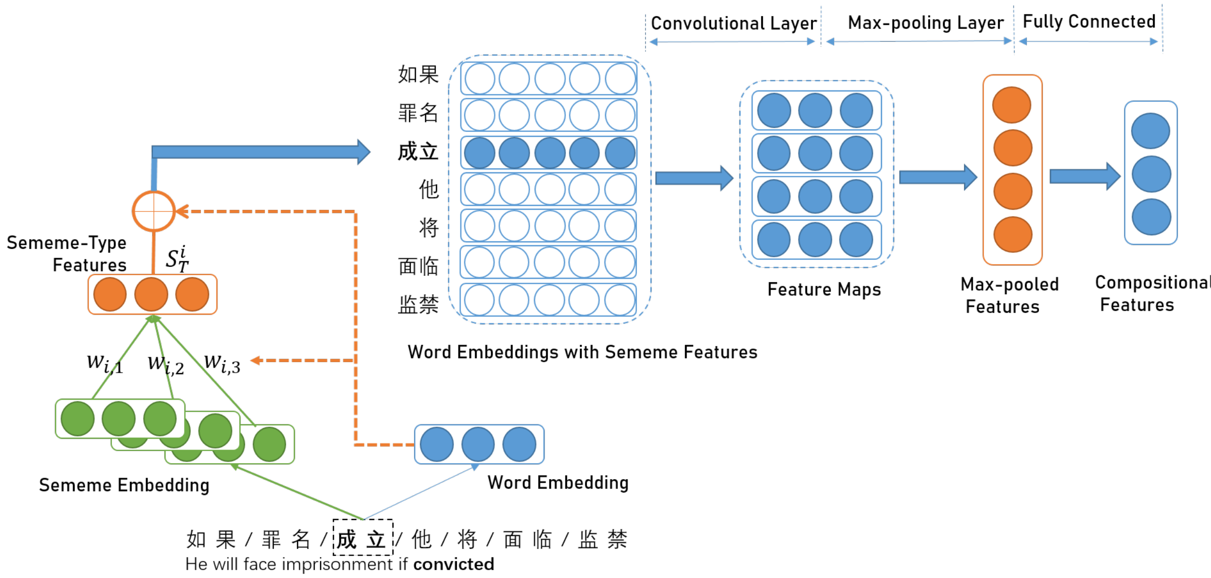


Fig. 4: The architecture of compositional representation learning with the features of word-sememe-type tertiary structure in word-level.

represents tanh function here. Therefore, we could obtain lexical representation in char-level as $\mathbf{r}_l$.

Compositional Stream Learning. The onefold compositional representation learning approach is similar with the procedure of character sequences. As the $\mathbf{r}_l$ generated by the learning approach, we conclude that lexical representation is consist of char-level embeddings learning from CNN and lexical embeddings fused by lexical gate. One significant difference between lexical and compositional stream learning is that word-level representation consolidates event sememe features. token embeddings learning for word is similar with the procedure of char-level representation learning and we skip the CNN part in order to avoid redundant expression.

Therefore, we describe the function of event sememe features incorporating with compositional features in word-level. Our event sememe information contributes to the model from two aspects. In the first part, for given sequence $S = \{x_0, x_1, ..., x_n\}$, we query each token whether it related to an event sememe. If so, we add the sememe embedding and finally we could get event sememe sequence $S' = \{s_1, s_2, ..., s_m\}$. We send the embeddings of S' to the CNN model, after the max-pooled layer we can get the most important features $\mathbf{r}_{es}$ for event sememes in this sentence.

As for the second part, we enrich word embeddings with word-sememe-type annotated information. As each word is defined by one or more sememes, and each sememe we selected is annotated with not only one event type. Assuming x_c is the concerning center word in S and s_i^c is the ith event sememe for x_c. Hence, the contribution of each sememe for its corresponding token is computed as following:

$$\mathbf{w}_i^c = \frac{exp(\mathbf{x}_c \cdot \mathbf{s}_i^c)}{\sum_{j=1}^{m_c} exp(\mathbf{x}_c \cdot \mathbf{s}_j^c)} \tag{5}$$

where m_c is the total number of sememes for x_c, $\mathbf{w}_i^c$ is the weight of ith sememe for x_c. Rely on the re-weighting value $\mathbf{w}^c$ we could obtain sememe-type embeddings for its current center word:

$$\mathbf{s}^c = \sum_{k=1}^{m_c} \mathbf{w}_k^c \odot \mathbf{s}_k^c \tag{6}$$

We illustrate the fusion method in Fig. 4. Then we concatenate sememe-type embeddings and word embeddings before we put them into convolutional layer to form structural features. After that, we get the feature $\mathbf{r}_{ws}$ for the next step. In order to integrate event sememe features in sentence-level, we unite both of them as $\mathbf{f}_C = \mathbf{r}_{ws} \oplus \mathbf{r}_{es}$ and employ a compositional gate to gain compositional representation:

$$\mathbf{r}_c = \sigma(\mathbf{W}_C \mathbf{f}_C + \mathbf{b}_C) \tag{7}$$

Therefore, we could obtain compositional representation as $\mathbf{r}_c$.

C. Hybrid Representation Fusion

By learning lexical stream in char-level and compositional stream in word-level separately, we could maximize the contribution of features in different aspects for both identifier and classifier. We could summarize from previous works that semantic clues in char-level help a lot for trigger identifier and structural clues in word-level play an significant role in type classifier rely on sememe knowledge.

Therefore, we learn representation for both identifier and classifier separately as follows:

$$\begin{aligned}\mathbf{f}_{iden} &= \boldsymbol{\alpha}_I^l \odot \mathbf{r}_l + \boldsymbol{\beta}_I^c \odot \mathbf{r}_c \\ \mathbf{f}_{cla} &= \boldsymbol{\alpha}_C^l \odot \mathbf{r}_l + \boldsymbol{\beta}_C^c \odot \mathbf{r}_c\end{aligned} \tag{8}$$

where $\boldsymbol{\alpha}^l$ and $\boldsymbol{\beta}^c$ are lexical gate vector and compositional gate vector which are learned separately for identifier and classifier. Until now, these two representation are prepared for the final stage.

D. Model Training

In this paper, the event detection task evaluation is divided into trigger word identifying score and event type classifying score. To adaptively map feature to trigger identifier and type classifier extracted from hybrid representation, we employ nonlinear function to map these features into its own space:

$$\mathbf{r}_i = \sigma_{d^I}(\mathbf{f}_{iden}); \mathbf{r}_c = \sigma_{d^C}(\mathbf{f}_{cla}) \tag{9}$$

where σ is linear layer with a nonlinear function, d^I is on behalf of the space dim for trigger identifier and d^C is the space dim for event type classifier. For a input sentence $s = \{x_1, x_2, ..., x_n\}$ we generate corresponding trigger word identifying label sequence as $y_i^I = \{y_1, y_2, .., y_I\}$ and event type classifying label sequence as $y_i^C = \{y_1, y_2, ..., y_C\}$ to train thses two classifiers. So the probability distribution computed stepwise for each instance is shown:

$$\begin{aligned}P(x, y_i^I; \theta) &= \frac{exp(\mathbf{r}_I^i)}{\sum_{j=1}^{S_T} exp(\mathbf{r}_I^j)} \\ P(x, y_i^C; \theta) &= \frac{exp(\mathbf{r}_C^i)}{\sum_{j=1}^{S_C} exp(\mathbf{r}_C^j)}\end{aligned} \tag{10}$$

Until now, we can obtain the trigger word identifier loss and event type classifier loss relying on cross entropy between the predictions and the ground-truth:

$$\begin{aligned}\mathcal{L}_{iden}(\theta) &= -\sum_{(x_m, y_m^I) \in S_I} \log P(x, y_m^I; \theta) \\ \mathcal{L}_{cla}(\theta) &= -\sum_{(x_m, y_m^C) \in S_C} \log P(x, y_m^C; \theta)\end{aligned} \tag{11}$$

and the total loss of our model is summarized as:

$$\mathcal{L}(\theta) = \mathcal{L}_{iden}(\theta) + \mathcal{L}_{cla}(\theta) \tag{12}$$

Due to continuously iteration of back-propagation in model training, our model would accurately identify the trigger word span in a sentence and make a rational prediction about the event type of trigger words.

IV. Experiments

A. Dataset and Evaluation Settings

Experiments of ED task is conducted on ACE2005 Chinese corpus which defines 8 event types in different directions and 33 event subtypes are divided into these 8 event types. It should be noted that the classification result of this paper is established on event subtypes ignoring the hierarchy between event types. In order to make an effective comparison with other baselines, we follow the same setup of dataset [8], [9], [28], [29] with 697 articles in total and 569 documents for training set, 64 for development set and the rest 64 used as testing set.

We use micro-averaged Precision(P),Recall(R) and F1-score as the evaluation metric for our model which is also same as [28]. The pre-trained character, word and sememe embedding size for representation learning are shown in Table II as well as the hyper-parameters used in our model. It should be noted that we apply AdamDelta method [30] for hyper-parameter optimization.

TABLE II: Hyper-parameters Setup In Our Model

Hyper-parameters	**Value**
char embedding size	200
word embedding size	200
trigger sememe embedding size	200
token position size	5
event type labeling size	34
cnn feature map size	400
representation fusion size	500
dropout rate	0.5
learning rate	1.0

B. Baselines and Overall Results

We make a comparison with previously proposed excellent approaches. We divide the baselines into three groups:

Char-based Approaches. This part focuses on extracting event trigger clues in a character-level to achieve object of task.

MEMM architecture [28] achieved competitive performance in char-level via neighbor word features extracting.

C-BiLSTM [12] utilizes convolutional Bi-LSTM model in char-level to capture both sentence-level and lexical features from context.

Word-based Approaches. Features in word-level show a great advantage in Chinese event extraction.

HNN [29] employs CNN and Bi-LSTM to build a hybrid neural network architecture.

C-BiLSTM model [12] also made an experiment on word-level features which is extracted by the same architecture as char-level.

Hybrid Representation Learning.This part is generally achieved via representation fusion from character features and word features, sentence-level and manually annotated features in some cases.

Rich-C [31] produced handcraft Chinese-specific features to deal with difficulties met in event extraction task.

NPN [8] proposed a comprehensive model to fuse inner compositional clues and ambiguous semantic clues in a particular way, which performs well in match accuracy of trigger words detection. And it is one of the state-of-thr-art architecture for our paper.

TABLE III: Overall Experiments Results on ACE2005[a]

Model		ACE2005					
		Trigger Identifier			*Type Classifier*		
		P	***R***	***F1***	***P***	***R***	***F1***
Char-based Model	MEMM* [28]	64.40	36.40	46.50	60.60	34.30	43.80
	C-BiLSTM+Errata table* [12]	53.00	52.20	52.60	47.30	46.60	46.90
Word-based Model	HNN* [29]	74.20	63.10	68.20	77.10	53.10	63.00
	C-BiLSTM+Errata table* [12]	56.50	47.00	51.30	49.60	41.30	45.00
Hybrid-based Model	Rich-C* [31]	62.20	71.90	66.70	58.90	68.10	63.20
	NPN(Task-specific)* [8]	64.80	**73.80**	69.00	60.90	**69.30**	64.80
Our Model	Lexical-only	66.54	64.41	65.46	62.90	60.89	61.88
	Concat-only	67.07	69.86	68.44	63.68	66.34	64.98
	Hybrid Representation	**81.11**	61.34	**69.86**	**78.89**	59.66	**67.94**

[a]Symbol * indicates the result adapted from the original paper.

As shown in Table III, the results conducted on ACE2005 in our experiment make a progress compared with the above baselines.

We train our model from three aspects so as to evaluate the performance of various methods. Lexical-only model indicates the effects about semantic clues captured from lexical stream learning method which considered character embeddings as initial token representation. We explain simple concatenation between lexical and compositional representation as the concat-only model, and the event sememe clues is computed into compositional stream. As for hybrid representation, we conduct word-sememe-type hierarchy structure information as re-weight value for compositional representation in word-level. From the results in Table III we can observe that:

Together with lexical and compositional representation our model achieves competitive performance in Chinese event detection. Compared with NPNs(task-specific) which is one of the best baseline for our paper, the hybrid representation learning model achieves 0.86(1.2%) and 3.14(4.8%) F1-score improvement on event trigger classification task on ACE2005 especially a great progress for type classification.

Compositional representation learning with event sememe information helps a lot for our model capturing event trigger clues in sentence-level. As trigger words in a sentence make a small quantity, we aad event sememe embeddings into neural network so that the compositional stearm could obtain more trigger candidates features and improves the accuracy of trigger classification. The evaluating results suggests that event sememes could increase the chance of trigger classification according to the F1-score promotions.

Word-sememe-type hierarchy structure reweights sememe words features in hybrid representation which effectively resolves trigger identification. By employing relation between each event sememe and event type annotated as tag labels, we can adjust the sensibility of trigger words identifying for model due to the event sememe information transfuse into hybrid representation. As the experiment results, some triggers with the cross-words shape or even being sparse in testing set could be detect toillessly depending on event sememe annotated information. And the type tag information gives the model event type clues to alleviate errors in trigger type classification.

C. Analysis of Mismatch and Sparse Labels

Detecting trigger words and event type labeled in ACE2005 Chinese corpus suffers a lot from language segmentation. Given a trigger token, if we identify the span of trigger candidate using lexical stream learning approach we can get the probability distribution for each character played roles of candidate within a fixed-length scale. From this method, no matter a candidate is made up of one or more characters our model could cover all cases like inside-word trigger, cross-word trigger and others. Such as the trigger candidate in Table IV. However, it brings trigger overlap problem which influence a lot in the performance of model. The feature extracted from training is not enough for model to observe the correct span for trigger and minimize the effect of negative example on the model.

As for the contributions of compositional representation learning for our model, it performs well in general trigger words like “当选”(be selected) triggers the event type “*Personnel:Elect*”, “牺牲”(sacrifice) triggers the event type “*Life:Die*” and “送”(send) for event type “*Movement:Transport*”. These kind of words could be detected as a single token to trigger corresponding event types and it shows friendly for model to capture word-level features. On the contrary, trigger words like “击毙”(shoot and kill) triggers two different type “*Conflict:Attack*” and “*Life:Die*”, and “一网打尽”(catch all) triggers one type “*Justice:Arrest-Jail*” which consist of four characters. Such kinds of trigger candidates confuse the model in trigger word span selection. Compositional representation learning approach could capture the structural clues, i.e., trigger word patterns in “蜂拥而来”(swarm forard) and “扬长而去”(swagger off) both have “adj.+ 而 +verb.” format and trigger the same event “*Movement:Transport*”. Summarized these structural features contributes to detect cross-word trigger candidates and mitigate the influence of mismatch problem in this task.

TABLE IV: Examples about Event Sememes with Annotated Tags Making Efforts On this Task[a]

Sentence	**Sememe Set**	**Annotated Type**	**Baseline**	**Our Model**
…公司\会\有**并购**\的\情形 …companise have **mergers** and **acquisitions**	(买,buy) (经济,economy)	(Transaction: Transfer-Ownership) (Business: Merge-Org)	(并购: Business:Merge-Org)	(并购: Business:Merge-Org) (并购: Transaction: Transfer-Ownership)
法官\随即**判**\被告… The judge then **sentenced** the defendant…	(裁定,adjudicate) (警,admonish)	Justice:Sentence Justice:Convict Justice:Arrest-Jail	(判: Justice:Convict)	(判: Justice:Sentence)
结果\谈判\不\成\而**大打出手**… But negotiations failed and **fought**…	(打,hit) (暴,violence)	Conflict:Attack	(打: Conflict:Attack)	(大打出手: Conflict:Attack)
…**当选**\美国\总统 …be **selected** as president of the US	(获胜,win) (选拔,select)	Personnel:Elect	None	(当选: Personnel:Elect)

[a]Word 'None' indicates the trigger type has been missed.

Sparsely labeled trigger words contains out-of-vocabularies and out-of-labels. These two types account for minor proportions in dataset which represents unseen or sparse triggers. Some candidates only appear in testing set or the labels of triggers are absent in training set so that the training model could not obtain features about them. Words like "狂轰滥炸"(bomb savagely) triggers event "*Conflict:Attack*" and "发短信"(send messages) is for event "*Contact:Phone-Write*" have not shown in the training process. As for "{中弹,*Life:Injure*}", "{成立,*Justice:Convict*}" and so on, the trigger words arise in training set but the label features could not be captured. Fusing both lexical and compositional representation with sememe knowledge does a favor of sparse trigger words detection.

D. Effect of Event Sememe Memory

TABLE V: Comparison with Three Kinds of Models on F1-scores[a]

Model	**TI-F1**	**TC-F1**
Modeling event-sememe only	67.67	66.76
Modeling tertiary structure only	69.37	66.06
Modeling hybrid approach	69.86	67.94

[a]TI is the abbreviation for trigger identifier and TC directs type classifier.

Application of event sememes is divided into two parts: event sememe embeddings and word-sememe-type consolidation. The first part operates separately with CNN model and then is fused into compositional representation as sememe trigger information in sentence-level. The latter one computes each sememe type as weight values to reset word-sememe embeddings with annotated information and then concatenates with word embeddings for the compositional feature learning.

The effectiveness of each method is shown in tableV. As the results suggested, modeling with event-sememes can help with improving the ability of trigger type classifier, where F1-score gains significant promotion compared with the second one. Event-sememe features provide information for event triggers in sentence-level so as to help model detect triggers with more event type clues. In the meanwhile, an obvious F1-score improvements for trigger identifier shown up in word-sememe-type tertiary structure fusing method. We summarize that re-weighting word-sememe embeddings with type information can help the model detect the probability of each trigger word for being a trigger candidates. By consolidating event-sememe embeddings and word-sememe-type annotated information, the compositional features is enriched with sememe trigger information. And our model could achieve event trigger detecting in both word-level and sentence-level features.

E. Case Study

In oder to describe the effectiveness of event sememes which is the external knowledge with sememe type annotated information, we illustrates several instances compared with the baseline in Table IV. Trigger "并购" (merge and acquisition) in the first sample has two different event types, word "并" triggers event type "*Business:Merge-Org*" and word "购" is for event type "*Transaction:Transfer-Ownership*". The baseline considers the whole word as a token and detects only one type, while our model detect two types for this trigger word cause of that the word is directed by its event sememes and the type has been annotated as a tag for its sememe-type embeddings. The same situation is analyzed in the second and third instances. The baseline gives trigger word "判" (sentence) the event type "*Justice:Convict*" in the second sentence. but we detect as "*Justice:Sentence*" because the word "判" (sentence) is related to two sememes and we annotate three type labels rely on the concept of sememes. And in the third instance, the rigger is considered as "打" (hit) for type "*Conflict:Attack*" but in our model the trigger word span is set as "大打出手" (fight).

Besides the type labels missed problem, error classification for event triggers and wrong token span detecting problems, the trigger detecting could be missed by traditional approach. Such as the forth instance, word "当选" (select) is related with event sememe "选拔"

(select) with event type labeled information. Hence our model detected “当选” (select) as trigger candidate with event sememe clues but the baseline missed this trigger candidate. Apparently, it is not flexible to classify triggers only with event sememe information cause of the error propagation for manual annotation. Thus it is necessary to fuse event sememe information and word representation to extract trigger word features, especially with the final hybrid representation.

V. Conclusion

In this paper, we employ CNN model to generate lexical representation which extracts semantic features of context in char-level, and compositional representation illustrating structural features in word-level. Besides, we utilize sememe knowledge in HowNet in order to enrich compositional clues. In detail, we select a set of event sememes with labeled event types for ED task, the event types information is annotated manually rely on the annotation rules of sememes and triggers which are related to trigger words. The usage of event sememes is divided into event sememe embeddings and word-sememe-type with annotated information The former part provides sememe trigger clue in sentence-level after the convolutional computing, and the latter one contributes to re-weighting compositional features for each word. According to experiment results, our model achieves competitive performance especially in event type classification.

As the limited quantity of trigger words in ACE2005 corpus, we would like to conduct the event sememe information on other corpus in the future, in order to achieve the general applicability of sememe knowledge. In addition, we attempt to design a more suitable architecture for word-sememe feature extracting to maximize the function of sememe information.

Acknowledgment

The work described in this paper was mainly supported by the grants from the National Natural Science Foundation of China (No.61672100), and Beijing Natural Science Foundation (No. 4202069).

References

[1] D. Ahn, “The stages of event extraction,” in *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*. Sydney, Australia: Association for Computational Linguistics, Jul. 2006, pp. 1–8. [Online]. Available: https://www.aclweb.org/anthology/W06-0901

[2] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, “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 (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 167–176. [Online]. Available: https://www.aclweb.org/anthology/P15-1017

[3] T. H. Nguyen, K. Cho, and R. Grishman, “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*, 2016.

[4] T. H. Nguyen and R. Grishman, “Graph convolutional networks with argument-aware pooling for event detection,” in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[5] X. Liu, Z. Luo, and H. Huang, “Jointly multiple events extraction via attention-based graph information aggregation,” *CoRR*, vol. abs/1809.09078, 2018. [Online]. Available: http://arxiv.org/abs/1809.09078

[6] S. Liu, R. Cheng, X. Yu, and X. Cheng, “Exploiting contextual information via dynamic memory network for event detection,” *CoRR*, vol. abs/1810.03449, 2018. [Online]. Available: http://arxiv.org/abs/1810.03449

[7] W. Lu and T. H. Nguyen, “Similar but not the same: Word sense disambiguation improves event detection via neural representation matching,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 4822–4828. [Online]. Available: https://www.aclweb.org/anthology/D18-1517

[8] H. Lin, Y. Lu, X. Han, and L. Sun, “Nugget proposal networks for chinese event detection,” *CoRR*, vol. abs/1805.00249, 2018. [Online]. Available: http://arxiv.org/abs/1805.00249

[9] N. Ding, Z. Li, Z. Liu, H. Zheng, and Z. Lin, “Event detection with trigger-aware lattice neural network,” 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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 347–356. [Online]. Available: https://www.aclweb.org/anthology/D19-1033

[10] Y. Niu, R. Xie, Z. Liu, and M. Sun, “Improved word representation learning with sememes,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 2049–2058. [Online]. Available: https://www.aclweb.org/anthology/P17-1187

[11] J. Liu, Y. Chen, K. Liu, and J. Zhao, “Event detection via gated multilingual attention mechanism,” 2018. [Online]. Available: https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16371

[12] Z. Ying, H. Yang, Y. Feng, W. Zheng, and D. Zhao, “A convolution bilstm neural network model for chinese event extraction,” in *International Conference on Computer Processing of Oriental Languages National CCF Conference on Natural Language Processing and Chinese Computing*, 2016.

[13] L. Sha, F. Qian, B. Chang, and Z. Sui, “Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction,” 2018. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16222/16

[14] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” *CoRR*, vol. abs/1802.05365, 2018. [Online]. Available: http://arxiv.org/abs/1802.05365

[15] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[16] S. Yang, D. Feng, L. Qiao, Z. Kan, and D. Li, “Exploring pre-trained language models for event extraction and generation,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 5284–5294. [Online]. Available: https://www.aclweb.org/anthology/P19-1522

[17] Z. Dong and Q. Dong, *HowNet - a hybrid language and knowledge resource*, Nov. 2003.

[18] Z.-D. Dong, Q. Dong, and C.-L. Hao, “Theoretical findings of hownet,” *Journal of Chinese Information Processing*, vol. 21, no. 4, pp. 3–9, 2007.

[19] P. Li and G. Zhou, “Employing morphological structures and sememes for Chinese event extraction,” in *Proceedings of COLING 2012*. Mumbai, India: The COLING 2012 Organizing Committee, Dec. 2012, pp. 1619–1634. [Online]. Available: https://www.aclweb.org/anthology/C12-1099

[20] F. Qi, J. Huang, C. Yang, Z. Liu, X. Chen, Q. Liu, and M. Sun, “Modeling semantic compositionality with

sememe knowledge," *CoRR*, vol. abs/1907.04744, 2019. [Online]. Available: http://arxiv.org/abs/1907.04744

[21] Z. Li, N. Ding, Z. Liu, H. Zheng, and Y. Shen, "Chinese relation extraction with multi-grained information and external linguistic knowledge," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4377–4386. [Online]. Available: https://www.aclweb.org/anthology/P19-1430

[22] L. Luo, X. Ao, Y. Song, J. Li, X. Yang, Q. He, and D. Yu, "Unsupervised neural aspect extraction with sememes," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, S. Kraus, Ed. ijcai.org, 2019, pp. 5123–5129. [Online]. Available: https://doi.org/10.24963/ijcai.2019/712

[23] L. Zhang, F. Qi, Z. Liu, Y. Wang, Q. Liu, and M. Sun, "Multi-channel reverse dictionary model," *ArXiv*, vol. abs/1912.08441, 2019.

[24] P. Li, G. Zhou, Q. Zhu, and L. Hou, "Employing compositional semantics and discourse consistency in chinese event extraction," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012.

[25] T. H. Nguyen and R. Grishman, "Modeling skip-grams for event detection with convolutional neural networks," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 886–891. [Online]. Available: https://www.aclweb.org/anthology/D16-1085

[26] ——, "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 (Volume 2: Short Papers)*. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 365–371. [Online]. Available: https://www.aclweb.org/anthology/P15-2060

[27] B. Yang, N. Li, Z. Lu, and J. Jiang, "Event detection with convolutional neural networks for forensic investigation," in *Intelligent Information Processing VIII*, Z. Shi, S. Vadera, and G. Li, Eds. Cham: Springer International Publishing, 2016, pp. 97–107.

[28] C. Zheng and H. Ji, "Language specific issue and feature exploration in chinese event extraction," in *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*, 2009.

[29] X. Feng, Q. Bing, and T. Liu, "A language-independent neural network for event detection," *Science China(Information Sciences)*, vol. v.61, no. 9, pp. 81–92, 2016.

[30] M. Zeiler, "Adadelta: An adaptive learning rate method," vol. 1212, Dec. 2012.

[31] C. Chen and V. Ng, "Joint modeling for Chinese event extraction with rich linguistic features," in *Proceedings of COLING 2012*. Mumbai, India: The COLING 2012 Organizing Committee, Dec. 2012, pp. 529–544. [Online]. Available: https://www.aclweb.org/anthology/C12-1033