Selective Expression For Event Coreference Resolution on Twitter

Wenhan Chao[†], Ping Wei[†], Zhunchen Luo[‡*] Xiao Liu[§], Guobin Sui[†]
[†]School of Computer Science and Engineering
Beihang University, Beijing, China 100191
Email: {chaowenhan, weiping, suigb}@buaa.edu.cn
[‡]Information Research Center of Military Science
PLA Academy of Military Science, Beijing, China 100142
Email: zhunchenluo@gmail.com
[§]School of Computer Science and Technology
Beijing Institute of Technology, Beijing, China 100081
Email: xiaoliu@bit.edu.cn

Abstract*—With the growth in popularity and size of social media, there is an urgent need for systems that can recognize the coreference relation between two event mentions in texts from social media. In existing event coreference resolution research, a rich set of linguistic features derived from pre-existing NLP tools and various knowledge bases is often required. This kind of methods restricts domain scalability and leads to the propagation of errors. In this paper, we present a novel selective expression approach based on event trigger to explore the coreferential relationship in high-volume Twitter texts. Firstly, we exploit a bidirectional Long Short Term Memory (Bi-LSTM) to extract the sentence level and mention level features. Then, to selectively express the essential parts of generated features, we apply a gate on sentence level features. Next, to integrate the time information of event mention pairs, we design an auxiliary feature based on triggers and time attributes of the two event mentions. Finally, all these features are concatenated and fed into a classifier to predict the binary coreference relationship between the event mention pair. To evaluate our method, we publish a new dataset EventCoreOnTweet (ECT)[1] that annotates the coreferential relationship between event mentions and event trigger of each event mention. The experimental results demonstrate that our approach achieves significant performance in the ECT dataset.

***Index Terms*—Event coreference resolution, Bi-directional long short term memory, Neural networks, Twitter**

I. Introduction

In recent years, social networks, like Facebook and Twitter, have become a popular platform for expressing opinions, broadcasting news and communicating with others. According to statistics, people usually post hundreds of tweets for significant events per day on Twitter. We need the ability to identify equivalent classes of event mentions to integrate and utilize information better from social platform like Twitter. Most previous approaches for solving event coreference mainly focus on corpora in which texts are written formally, such as MUC, ACE 2005[2], ECB [5], OntoNotes and Intelligence Community (IC) [6]. However, we are facing great challenge because of the emoticons, URLs, spam texts and misspellings in those social text such as tweets.

The task of event coreference resolution aims at identifying clusters of event mentions that event mentions in the same cluster refer to the same unique event in the real world. Event coreference resolution can be applied within-document and cross-document and is widely used in many natural language processing tasks including topic detection and tracking [1], information extraction [2], question answering [3] and semantic similarity computation [4].

Traditional approaches for event coreference resolution utilize rich semantic features derived from various linguistic knowledge bases and external sources, which are often extracted from the output of existing natural language processing system, not only restricting domain scalability but also may lead to the propagation of errors [43]. To reduce the propagated errors as much as possible and increase the system's scalability, we need to consider a new way to learn event features. In information extraction field, related problems such as relation classification [10], event extraction/detection [11], [12] and event linking [13] has achieved better performances than traditional models by utilizing neural network methods. Their studies inspire and attract our work to process event coreference resolution on Twitter to generate latent features by taking event trigger and its context into account from neural networks.

In contrast to entity coreference resolution, which connects and groups entity mention refer to the same discourse entity, event coreference resolution has not been extensively studied. Prior entity coreference resolution works [7], [8] result in better performance by incorporating the verbs in context and looking for the semantic role as features. Delip Rao [9] proposes a streaming clustering algorithm to identify coreferential entity mentions on Twitter, but these approaches cannot handle event coreference resolution, because it has more complex semantic representation and more flexible linguistic structures.

To address these limitations, we design a novel model for event coreference resolution on Twitter. Firstly, our model

[*] Corresponding author.

[1] The ECT dataset is available at https://github.com/pinggeger/ECT

[2] http://projects.ldc.upenn.edu/ace/

uses bidirectional Long Short Term Memory (Bi-LSTM) to generate sentence level features for tweet text and mention level features for event mention. Secondly, we notice that each word plays a distinct role in different event trigger in the same sentence, such as core word and confusing word. Thus, we employ a selective gate structure to filter the semantic expression of unimportant or irrelevant words according to event trigger from sentence level features. Next, we use the attention mechanism [32], [33] to recombine the expression of each sentential features with different weights to generate latent features. Finally, we concatenate latent features, mention features, and local features to make our coreferential decision. To investigate the effectiveness of our purposed approach, we created a new corpus EventCoreOnTweet (ECT) which contains event coreference annotations (coreference index and event trigger) from the Twitter Streaming API which offers an approximately 1% sample of all tweets. Experiments show the effectiveness of our selective expression model for event coreference resolution.

To sum up, our contributions are: (1) we propose a novel selective expression approach for event coreference resolution on Twitter. (2) to evaluate our approach, we publish a tweet dataset EventCoreOnTweet (ECT) that annotates the coreferential relationship between event mentions and event trigger of each event mention. (3) the experimental results demonstrate that our approach achieves significant performance in the ECT dataset.

II. Related Work

Clustering. Compared with entity coreference, event coreference resolution is much less active in research. Early works [2], [14] propose a general approach for event coreference on scenario specific events such as “elections”, “espionage” and “resignations”. Chen et al. [15] formally state the problem of event coreference resolution in the ACE [3] program, formulate an agglomerative clustering algorithm for this task and explore the feature impacts in the event coreference model. Chen and Ji [16] model event coreference resolution as a spectral graph clustering problem and evaluate the clustering algorithm on ground truth event mentions. Liu et al. [17] train the classifier by random forest and cluster the processes of all pairwise scores to decide the final clusters of each mention pairs. Peng et al. [18] purpose a semantic relatedness function to detect events and perform event coreference inference via a left-linking greedy algorithm.

Bayesian. Bejan and Harabagiu [5] notice that previous supervised learning methods [2], [16] rely on various in-domain linguistic properties and require a substantial amount of manual effort to annotate data. They propose two generative nonparametric Bayesian models based on hierarchical Dirichlet process [19] and infinite factorial hidden Markov model [20] for unsupervised within-document and cross-document event coreference resolution. Yang et al. [21] present a hierarchical distance dependent Bayesian model which leverage the advantage of supervised and unsupervised resolutions by encoding available supervisory information to guide the generative model toward better performance. Their model extends the framework of the distance-dependent Chinese restaurant process (DDCRP) [22] to make use of clustering priors with feature-based, learnable distance functions.

Regression. Most of the prior studies focus on entity coreference resolution, whereas Lee et al. [23] consider that noun phrases (NPs) could be events on entity coreference resolution. Moreover, events are characterized by triggers and arguments, which often correspond with discourse entities. Therefore, they revise and complete the EventCoreBank(ECB) corpus created by Bejan and Harabagiu [5] followed the OntoNotes [24] standard, and use a linear regression model to address reference to both entities and events across documents jointly. Araki et al. [25] notice that relationship detecting between event and sub-event can reduce the difficulty of full coreference resolution, so they introduce a multi-class logistic regression model to find and improve the hierarchical sub-event structure in addition to processing full event coreference resolution.

Neural Networks. Decisions based on rich semantic features from various knowledge bases not only restrict domains but also propagate errors from higher stream components. Krause et al. [13] propose a model to solve this problem by utilizing sentential features from convolutional neural networks instead of rich semantic features. Firstly, their model generates latent-feature representations by processed coreference candidates and their respective context. Then it concatenates lexical-level and pairwise features as input to a trainable similarity function to get the coreference score. Inspired by their work, we propose our selective expression approach to process event coreference resolution on Twitter.

III. Problem Formulation

Event coreference resolution on Twitter aims at identifying the coreferential relationship between two event mentions. An event mention consists of an event trigger, some participants, and time and location attributes. But an event trigger is not necessarily continuous and is restricted to nouns, noun phrases, verbs and verb phrases. We consider the coreference decision between two event mentions as a binary classification task whether the two event mentions are coreferential or not. Our model receives two tweets (context of event mention) T1, T2 and their event trigger as input and outputs the coreferential relationship between T1 and T2.

Here let us show a typical example, for the following two tweets *T1* and *T2*:

T1: @RedPillTweets Hillary’s team is paying crazy people to attack Trump supporters!, https://t.co/nsG8Ay6oHB

T2: @MSNBC undercover journalists expose DNC paid agitators to cause violence @ trump rallies. video evidence https://t.co/ ZumXQQamMi

We can see there are three event mentions in the two tweets *T1* and *T2* as below. (Event trigger for event mention is marked in bold, XXX is the context):

[3] http://www.nist.gov/speech/tests/ace/

EM1: XXX **is paying** *XXX* **to attack** *XXX*
EM2: XXX **expose** *XXX*
EM3: XXX **paid** *XXX* **to cause** *XXX*

In this typical example, the ground truth here is that EM1 and EM3 are coreferential, EM1 and EM2 are not coreferential as well as EM2 and EM3. In other words, EM2 is a singleton reference in this example.

IV. Model

In this section, we will introduce the details about our model. Our pairwise model consists of two parts, one for generating the semantic representation of event mention with recognized event trigger (Figure 1), and the other for making the coreferential decision between two event mentions (Figure 2).

A. Sentence and Mention Level Features

Tweets usually include informal writing, such as misspelling, grammatical errors, optional abbreviations, URLs, Hashtag, @someone, etc. To better process tweets, we need to do some tweets-oriented preprocessing, including segmentation, filtering URLs and emoticon and so on.

In our model, each segmented token in a tweet will be transform into a embedding vector. The embedding vector consists of two parts: word embedding and distance embedding. We use word2vec[4] [28] to pre-train word embedding, and apply the label "$\langle unknown \rangle$" to represent the out of vocabulary (OOV) words. Moreover, words that are close to the event trigger usually have a more relevant semantic relationship. Thus we use distance embedding to indicate the relative distance between words and event trigger. The two embedding matrices are then fine-tuned in training procedure.

Although vanilla RNNs can learn long dependencies in theory, in practice they fail to do so and tend to be biased towards their most recent inputs in the sequence. Long Short-term Memory Networks (LSTMs) [29] have been designed to combat this issue by incorporating a memory-cell and have been shown to capture long-range dependencies. They do so using several gates that control the proportion of the input to give to the memory cell, the proportion from the previous state to forget, and the proportion to output. They work well on a large variety of tasks [30], [31]. Thus we use LSTMs to encode the semantic representation of each word in their context. The detail of LSTM (1-6) is defined as the following formulas where W_f, W_i, W_c and W_o are weight matrics. "$*$" represents the multiply operation and "." represents the dot-multiply operation:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\widetilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * tanh(C_t) \tag{6}$$

For a given tweet, each token represented as an embedding vector, an LSTM $LSTM_f$ computes representations of the left context of the tweet at every token. $h_{f,i}$ is the hidden output of $LSTM_f$ cell when a token's embedding x_i is served as input. Naturally, generating a representation of the right context by a backward LSTM $LSTM_b$ as well should add useful information. This can be achieved by reading the same token sequence of the tweet in reverse. The two LSTMs are called one bidirectional LSTM (Bi-LSTM) (7-9), and we apply it to learn forward and backward sentence level and mention level features and then concatenate together. Finally, h_i is the concatenated context feature of i-th token.

$$h_{f,i} = LSTM_f[x_i, h_{f,i-1}] \tag{7}$$

$$h_{b,i} = LSTM_b[x_i, h_{b,i+1}] \tag{8}$$

$$h_i = [h_{f,i}, h_{b,i}] \tag{9}$$

After computing the context feature of each token, we can generate the sentence level feature and the mention level feature. The sentence level feature (dashed rectangle in the left of Figure 1) is a sequence of each word's context feature in tweet when tweet as input (Figure 1) where n is the length of tweet.

$$Sent_{level} = (h_0, h_1, h_2, ..., h_n) \tag{10}$$

And the mention level feature (dashed rectangle in the right of Figure 1) is the concatenation of the last output of forward and backward LSTM cell when event trigger as input (Figure 1) where m is the length of event trigger.

$$Ment_{level} = [h_{f,m}, h_{b,0}] \tag{11}$$

B. selective expression

We notice that each word plays a different role for one specific event trigger in the same sentence. Some are core words, while another portion of words is semantically confusing. Let us come back to the example T2 in section 3, there are two event mentions, EM2 and EM3, in T2. The deletion of words like "@MSNBC" and "video evidence" in EM2 and "@MSNBC undercover journalists expose" in EM3 does not affect the semantic expression of event mentions. However, some core words in EM2, such as "undercover journalists expose", might bring semantic confusion to EM3, and even wrongly induce the model to identify the coreferential relationship between EM2 and EM3. Therefore, we employ selective expression mechanisms to achieve more accurate latent features of event mentions by limiting the semantic expression of unimportant or irrelevant words according to the event trigger.

$$R_c = h_i * Ment_{level} \tag{12}$$

$$\alpha_i = tanh(W_s \cdot R_c + b_s) \tag{13}$$

In (13), W_s is the weight matrices, α_i is the weight parameter of the selective gate to indicate the quantity of i-th word's sentential features which can pass in tweet. For a tweet with n words, the weight parameter is $\alpha = (\alpha_0, \alpha_1, ..., \alpha_n)$,

[4] https://code.google.com/p/word2vec/

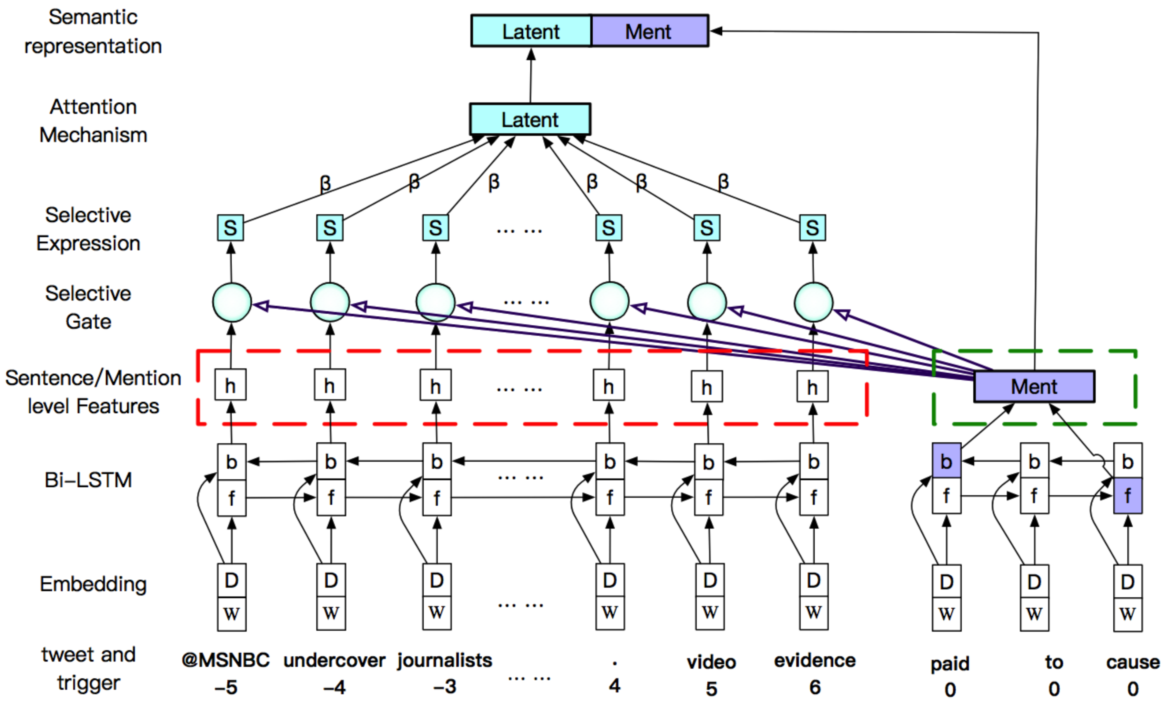

Fig. 1. Generating the semantic representation for a single event mention on Twitter

and the selective representation is formed by the product of α and $Sent_{level}$ as:

$$Select = \alpha * Sent_{level} \tag{14}$$

or

$$Select_i = \alpha_i * h_i \tag{15}$$

$$Select = (Select_0, Select_1, ..., Select_n) \tag{16}$$

C. Attention Mechanism

As we can observe that, a tweet with event mentions contains event triggers, participants, time, locations, and other entities. When a tweet contains multiple event mentions, the semantic relationship between words becomes more complicated. If there are different event mentions in a tweet, the semantic contribution of each word in the tweet to an event mention's semantic representation is varied. We use attention mechanisms to compute the importance score for each word's selective expression to reflect the different contributions to semantic representation of each event mention. Moreover, then we normalize the importance scores to obtain the latent feature by the weighted sum (17-19):

$$u_i = V_a^T tanh(W_a Select_i + b_a) \tag{17}$$

$$\beta_i = \frac{e^{u_i}}{\sum_{i=1}^{n} e^{u_i}} \tag{18}$$

$$latent = \sum_{i=1}^{n} \beta_i Select_i \tag{19}$$

In (17), W_a, b_a is the weight matrices and bias of attention layer. We concatenate the previous mention level feature and latent feature to generate the semantic features of a tweet as follows:

$$V_{em} = (latent, Ment_{level}) \tag{20}$$

D. Coreference Decision

Event coreference solution is the task of determining the coreferential relationship between two event mentions. The second part of our pairwise model (Figure 2) not only receives the semantic representations for two event mentions V_{em}^1, V_{em}^2 as input, but also rely on an important feature $V_{local}^{1,2}$. To our knowledge, all events on Twitter occurred at a specific time because the timestamp of posting every tweet will be always recorded. While a topical event would be discussed for a long time, the semantic meanings of event triggers usually belong to one or more similar limited types of events. To argument the time and semantic meaning similarity information and to enhance the semantic expressions of event mentions, we exploit the following pairwise features as local feature:

- V_W: The number of words in the overlap between event triggers.
- V_D: The number of days between two event mention. (Time of event mention equals to the time of the tweet posted by users to Twitter service.)

We can indicate our pairwise features as:

$$V_{pair} = (V_{em}^1, V_{em}^2, V_{local}^{1,2}) \tag{21}$$

$$V_{local}^{1,2} = (V_W, V_D) \tag{22}$$

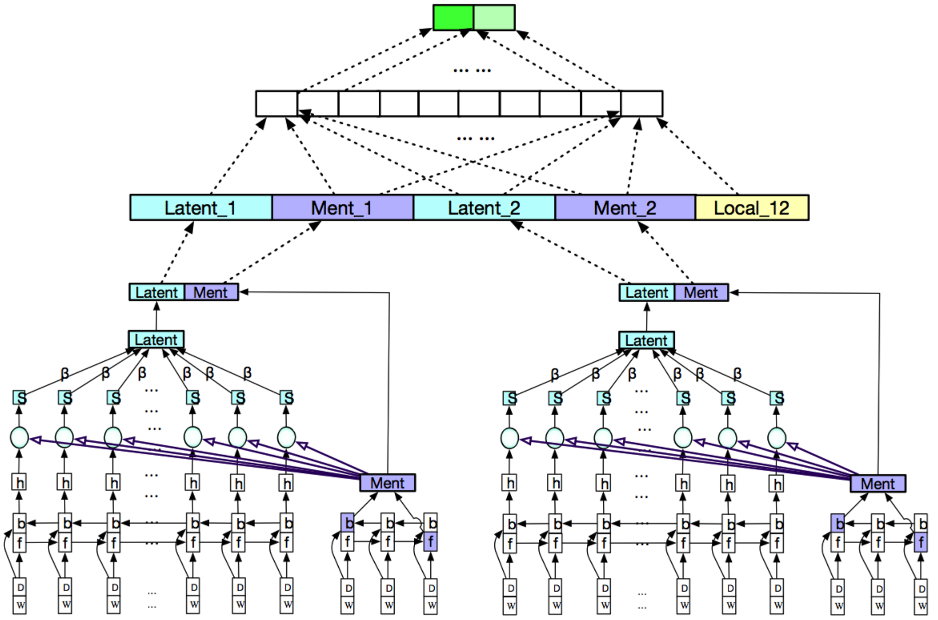

Fig. 2. Make coreferential decision of two event mention

And then the pairwise features are processed by a simple neural network layer to calculate the distributed similarity:

$$V_{ds} = relu(W_{ds} \cdot V_{pair} + b_{ds}) \tag{23}$$

We design our coreference decision as a binary classification model which has two categories, coreferential and not coreferential. Finally, we apply a softmax layer to calculate the probability of two categories:

$$Score = Softmax(W_{pro} \cdot V_{ds} + b_{pro}) \tag{24}$$

We learn our model parameters by minimizing the cross entropy (logistic) loss with Adam [42].

V. Experiments

A. Dataset

The Twitter streaming API[5] offers 1% sample of all tweets. We collected 10 million English tweets after filtering out retweets and tweets with no more than 10 words from the streaming API over a 31-day period. With U.S. Presidential election receiving high degree of attention among Twitter users, we attempt to focus on the coreferential relationship between events taking place during this period. We set the keywords ("trump", "hillary", "presidential election") to filter spam and uncorrelated tweets. It's impractical for us to annotate all the event mentions in the dataset, so we perform `twitter_nlp`[6] [34] to extract all triggers and ranking by frequency after stemming, finally we obtain 1.2 thousand tweets which include one of 20 words we chose from medium frequency triggers. Table I lists statistics for the dataset.

	Total
Tweet	2994
Event	1111
Event Mention	3879

TABLE I
STATISTICS FOR THE ECT DATASET

Annotators are firstly required to judge if the tweet mentions a event or not. And then for those tweets which mention an event, they need to annotate the event trigger and coreferential index. We employ Cohens Kappa [35] to measure the inter-annotator agreement between annotators. In judging event mention, our two annotators reached Cohens Kappa of 0.78. In annotation, our annotators reached Cohens Kappa of 0.84. We selected the consistent 2990 tweets as the golden standard corpus (EventCoreOnTweet, ECT). The following lists our annotation standard:

- We only focus on those were written in English.
- Retweets were not to be taken into account. As we mentioned before, we filter out RT tweets.
- The tweet must explicitly mention a event and the reader can infer what happened without any outside knowledge after reading the tweet.
- We only focus on those which are produced in the declarative way.
- We ignore information indicated by links. Finding information from URLs could be an interesting avenue for future work.
- We did not limit the number of event mentions in a tweet, which means a tweet can include one or more event mentions.
- Event trigger could be nouns, noun phrases, verbs and

[5] https://dev.twitter.com/docs/streaming-api
[6] https://github.com/aritter/twitter_nlp

verb phrases.

B. *Example generation and clustering*

```
Algorithm 1 GeneratePairExamples
$M_t$ = ($m_1$, $m_2$, $m_3$, ..., $m_{|M_t|}$), sorted by time
$P_m \leftarrow \emptyset$
for i = 1, ...., $|M_t|$ do
    for j = i, ...., $|M_t|$ do
        if Days($m_i$, $m_j$) $\leq$ Window then
            $P_m \leftarrow (m_i, m_j)$
return $P_m$
```

Fig. 3. Algorithm of generating examples.

We refer to the generation of examples from documents with recognized event mentions to create our examples on Twitter, which chronologically iterates over the event mentions in dataset and pairs each mention with all preceding ones within a fixed moving window. For example, if we have a anaphoric mention on Friday with a window of 7 days, the antecedent candidates must be these mention since last Friday. The detailed strategy is in Figure 3.

We adopt a way like transitive closure similar to Krause et al. [13] to induce the event clusters from all coreferential relationship from our model's outputs.

C. *Evaluation Metric*

We report results in terms of Precision (P), Recall (R) and F_1-score (F_1) using commonly-used coreference scoring algorithms given by the CoNLL[7] scorer to evaluate our event coreference resolution system. The scorer computes MUC [36] , B^3 [39], BLANC [37] and $CEAF_e$ [38]. We also report the CoNLL average [40], which is the average of MUC F_1, B^3 F_1, and $CEAF_e$ F_1.

D. *Experimental Setting*

We implement our model using the TensorFlow framework v1.2[8]. Our dataset ECT is split 9:1 into a development (dev) and test (test) partition, we further split the development partition 9:1 into a training (train) and validation (valid) partition. Table II lists statistics for the dataset.

TABLE II
STATISTICS FOR SPLITTING THE ECT DATASET

	Train	Vali	Test	Total
Tweet	2425	269	300	2994
Event	-	-	-	1111
Event Mention	3159	346	374	3879

We employ two baselines to examine the approach we proposed, one is our model without using selective gate, attention mechanism and local features, the other is Krause's [13] system, which we need to do some necessary changes to adapt it to twitter text. We removed three following features from pairwise because these features can not be applicable to tweets in our dataset.

- Agreement in event type
- Agreement in event modality
- Antecedent event is in first sentence

Besides, we replaced the bagged distance with the tweet's release interval V_D and the overlap in arguments with the overlap in event trigger. Other features and model parameters remained the same.

We set up different down-sampling and up-sampling ratio to conduct multiple experiments, and report the average over the all results. The final settings we used for all following experiments are listed in Table III.

TABLE III
HYPERPARAMETER SETTINGS

Name	Value
batch size	128
rnn size	128
attention size	128
word embedding size	100
distance embedding size	14
learning rate	0.01
decay rate	0.9
dropout	no

E. *Experimental Results*

Row 1 and row 2 of table IV show that the results of our baseline. Our RNN model improved the CNN model by 9.6 BLANC F1 points, 12.2 B^3 F_1 points, 11.3 $CEAF_e$ F_1 points, 6.6 MUC F_1 points and 10.0 CoNLL score points. We can see, RNN can accurately realize the semantic expression between event trigger and its context. Row 3 to row 5 show our model's performance when only using one of the three features we proposed based on RNN baseline, and row 6 to row 9 report our results when using different combinations of the three features. The last row of table IV corresponding to the full model as described in Section 4 achieves a better performance than two baselines by CoNLL score of 23.7 points (CNN) and by CoNLL score of 13.7 points (RNN). The following will analyze the performance of selective gate, attention mechanism, and local features.

Row 2 and row 3 indicate that the selective gate achieve a significant rise of 6.7 points on CoNLL score than the RNN baseline, the growth of which can also be verified in the comparative experiment row 8 and row 9, row 4 and row 6. These significant improvements indicate that the proposed selection gate structure can indeed filter out the semantic expression of unrelated words to guide our model to accurately generate the deep semantics of event mentions. Although there is a slight decline in the MUC F_1, careful observation of the precision and recall rate can reveal that the use of the selection gate structure results in a more balanced performance between them. Those results suggest that potential features

[7] https://github.com/conll/reference-coreference-scorers

[8] https://github.com/tensorflow/tensorflow

TABLE IV
EVENT COREFERENCE RESOLUTION PERFORMANCE OF OUR MODEL & COMPETITORS ON ECT

Approach	BLANC			BCUB			CEAFe			MUC			CoNLL
	P	R	F	P	R	F	P	R	F	P	R	F	F
CNN	54.4	65.3	52.8	33.3	77.7	45.5	65.9	26.2	33.2	53.4	72.7	61.1	46.6
RNN	59.2	72.7	62.4	44.8	81.1	57.7	68.1	33.0	44.5	57.7	82.0	67.7	56.6
RNN+G	72.6	79.0	**75.3**	62.2	76.1	68.4	64.2	50.3	56.4	60.4	70.7	65.2	63.3
RNN+A	64.3	70.1	66.6	55.7	75.6	64.1	71.6	51.1	59.6	61.1	75.1	67.4	63.7
RNN+L	66.2	71.2	68.2	55.4	75.5	63.9	74.4	51.4	60.8	60.5	75.6	67.2	64.0
RNN+G+A	65.9	**87.7**	71.7	58.0	**85.5**	69.1	73.7	51.7	60.8	66.1	82.0	73.2	67.7
RNN+G+L	69.7	73.3	71.3	59.4	75.8	66.6	68.9	51.2	58.8	61.5	74.1	67.3	64.2
RNN+A+L	**72.8**	73.2	73.0	**64.0**	77.7	70.2	75.8	**56.3**	64.6	64.8	78.0	70.8	68.5
RNN+G+A+L	68.4	85.1	73.7	60.7	85.4	**71.0**	**78.2**	55.4	**64.8**	**68.0**	**83.9**	**75.1**	**70.3**

"CNN" donates the system proposed by Krause et al.,
"RNN" donates to our system since we obtain features from LSTM.
"G", abbreviation of "Selective Gate", donates the selective gate to achieve selective expression;
"A", abbreviation of "Attention Mechanism" donates the attention mechanism;
"L" abbreviation of "Local Features" donates the local features.

extracted by the selective gate refine our model to achieve better performance.

From row 2 and row 4, we can see that attention mechanism outperforms the RNN baseline by 7.1 CoNLL score. It shows that the attention mechanism integrates the semantic features of each word and generates potential features which could better represent the event mention to improve the performance of the model. Although there was a slight decrease in MUC F_1, we observed that the difference between precision and recall of the model with attention mechanism was smaller than the RNN baseline model, and the use of the attention mechanism enhance robustness of our model. Moreover, the comparison between row 7 and row 9 demonstrates that giving a different weight to each component can better represent the semantics of event mentions.

The individual use of local features achieves an improvement of 7.4 points on CoNLL score than the RNN baseline. It shows that the addition of local features enriches the semantic differences between two event mentions and makes the model obtain higher resolving ability. The combination of attention mechanism and local features also bring better performance of 11.9 points of CoNLL score. These results provide us with an insight that local features play an important role, in large part it is because people were more inclined to use the same words to retell the event or to express their opinions when they saw a report of specific event, thus, the difference between triggers of event mentions is small in the same event, bigger in the different events. However, the combination of local features and selective gate (row 7) do not produce much improvement in our experimental results than only using local features (row 5), it arguably suggests that the latent features filtered by the selection gate need to be reassembled by attention mechanism to achieve better semantic representation. The full model obtains 70.3 CoNLL score (row 9) demonstrates that our approach achieves significant performance in feature extraction and expression.

VI. CONCLUSION AND FUTURE WORK

In this paper, we present a novel selective expression approach for event coreference resolution on Twitter, which limits the semantic expression of unimportant or irrelevant words according to the event trigger. We also create a tweet dataset EventCoreOnTweet (ECT) that annotates the coreferential relationship between event mentions and the event trigger of each event mention. Comparing with Krause's model and RNN model without using the selective gate, attention mechanism, and local features, we achieve a significant performance improvement over 23.7 CoNLL F_1 points and 13.7 CoNLL F_1 points. The result shows that the selective expression model we proposed achieves a significant performance of feature extraction and expression.

There are several directions we can focus on for the potential future work. On the one hand, we can design a structure of automatic recognition event trigger to complete an end-to-end event coreference resolution model on Twitter. On the other hand, with the background of big data, it is meaningful but challenging to quickly retrieve events from vast amounts of data in the social network, which allows us to extend our model to the streaming processing model further. Moreover, we need to explore if our model can be improved by extracting more fine-grained information, such as entity coreference resolution, the similarity of text indicated by URLs and user information.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful comments. We also thank our annotators for accomplishing the dataset and holding helpful discussions. This work is supported by National Key Research and Development Program of China (Grant No. 2017YFB1402400).

REFERENCES

[1] Allan J, Carbonell J, Doddington G, et al. Topic Detection and Tracking Pilot Study Final Report[C]// Darpa Broadcast News Transcription and Understanding Workshop. 1998:194–218.

[2] Humphreys K, Gaizauskas R, Azzam S. Event coreference for information extraction[C]// A Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts. Association for Computational Linguistics, 1997:75-81.

[3] Tellex S, Katz B, Lin J, et al. Quantitative evaluation of passage retrieval algorithms for question answering[C]// International ACM SIGIR Conference on Research and Development in Informaion Retrieval. ACM, 2003:41-47.

[4] Mccarthy D, Carroll J. Disambiguating Nouns, Verbs, and Adjectives Using Automatically Acquired Selectional Preferences[J]. Computational Linguistics, 2003, 29(4):pgs. 639-654.

[5] Bejan C A, Harabagiu S. Unsupervised event coreference resolution with rich linguistic features[C]// Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2010:1412-1422.

[6] Hovy E, Mitamura T, Verdejo F, et al. Events are not simple: Identity, non-identity, and quasi-identity[C]//NAACL HLT. 2013, 2013: 21.

[7] Haghighi A, Dan K. Coreference resolution in a modular, entity-centered model[C]// Human Language Technologies: the 2010 Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2010:385-393.

[8] Rahman A, Ng V. Coreference Resolution with World Knowledge.[C]// The, Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, Usa. DBLP, 2011:814-824.

[9] Rao D, Mcnamee P, Dredze M. Streaming Cross Document Entity Coreference Resolution[C]// COLING 2010, International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China. DBLP, 2010:1050-1058.

[10] Zeng D, Liu K, Lai S, et al. Relation classification via convolutional deep neural network[J]. 2014.

[11] Nguyen T H, Grishman R. Event Detection and Domain Adaptation with Convolutional Neural Networks[C]//ACL (2). 2015: 365-371.

[12] Chen Y, Xu L, Liu K, et al. Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks[C]// The, Meeting of the Association for Computational Linguistics. 2015.

[13] Krause S, Xu F, Uszkoreit H, et al. Event Linking with Sentential Features from Convolutional Neural Networks[C]// Signll Conference on Computational Natural Language Learning. 2016:239-249.

[14] Bagga A, Baldwin B. Cross-document coreference: Annotations, Experiments, and Observations[J]. In Proc. ACL-99 Workshop on Coreference and Its Applications, 1999, 1:1–8.

[15] Chen Z, Ji H, Haralick R. A pairwise event coreference model, feature impact and evaluation for event coreference resolution[C]// The Workshop on Events in Emerging Text Types. Association for Computational Linguistics, 2009:17-22.

[16] Chen Z, Ji H. Graph-based event coreference resolution[C]// The Workshop on Graph-Based Methods for Natural Language Processing. Association for Computational Linguistics, 2009:54-57.

[17] Liu Z, Araki J, Hovy E, et al. Supervised within-document event coreference using information propagation[J]. 2014.

[18] Peng H, Song Y, Dan R. Event Detection and Co-reference with Minimal Supervision[C]// Conference on Empirical Methods in Natural Language Processing. 2016:392-402.

[19] Teh Y W, Jordan M I, Beal M J, et al. Hierarchical Dirichlet Processes[J]. Publications of the American Statistical Association, 2006, 101(476):1566-1581.

[20] Gael J V, Teh Y W, Ghahramani Z. The infinite factorial hidden Markov model[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2008:1697-1704.

[21] Yang B, Cardie C, Frazier P. A Hierarchical Distance-dependent Bayesian Model for Event Coreference Resolution[J]. Computer Science, 2015.

[22] Blei D M, Frazier P I. Distance Dependent Chinese Restaurant Processes[J]. Journal of Machine Learning Research, 2011, 12(1):2461-2488.

[23] Lee H, Recasens M, Chang A, et al. Joint entity and event coreference resolution across documents[C]// 2012:489-500.

[24] Pradhan S S, Ramshaw L, Weischedel R, et al. Unrestricted Coreference: Identifying Entities and Events in OntoNotes[C]// International Conference on Semantic Computing. IEEE Computer Society, 2007:446-453.

[25] Araki J, Liu Z, Hovy E, et al. Detecting Subevent Structure for Event Coreference Resolution[J]. 2014.

[26] Zhang T, Li H, Ji H, et al. Cross-document Event Coreference Resolution based on Cross-media Features[C]// Conference on Empirical Methods in Natural Language Processing. 2017:201-206.

[27] Karami S, Boffetta P, Rothman N, et al. A Structured Distributional Semantic Model: Integrating Structure with Semantics[J]. Carcinogenesis, 2013, 29(8):20-29.

[28] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2013:3111-3119.

[29] Hochreiter S, Schmidhuber J. Long short-term memory.[J]. Neural Computation, 1997, 9(8):1735-1780.

[30] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[J]. 2014, 4:3104-3112.

[31] Wu Y, Schuster M, Chen Z, et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation[J]. 2016.

[32] Mnih V, Heess N, Graves A, et al. Recurrent models of visual attention[J]. 2014, 3:2204-2212.

[33] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. Computer Science, 2014.

[34] Ritter A, Mausam, Etzioni O, et al. Open domain event extraction from twitter[C]// Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012:1104-1112.

[35] Cohen J. A coefficient of agreement for nominal scales.[J]. Educational & Psychological Measurement, 2016, 20(1):37-46.

[36] Vilain M, Burger J, Aberdeen J, et al. A Model-Theoretic Coreferenc e Scoring Scheme[C]// Conference on Message Understanding, Muc 1995, Columbia, Maryland, Usa, November. DBLP, 1995:45-52.

[37] M. RECASENS, E. HOVY. BLANC: Implementing the Rand index for coreference evaluation[J]. Natural Language Engineering, 2011, 17(4):485-510.

[38] Luo X. On coreference resolution performance metrics[C]// HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada. DBLP, 2005:25–32.

[39] Bagga A, Baldwin B. Algorithms for Scoring Coreference Chains[J]. 1998, 5:563–566.

[40] Pradhan S, Luo X, Recasens M, et al. Scoring Coreference Partitions of Predicted Mentions: A Reference Implementation[C]// Meeting of the Association for Computational Linguistics. 2014:30-35.

[41] umphreys K, Gaizauskas R, Azzam S. Event coreference for information extraction[C]// A Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts. Association for Computational Linguistics, 1997:75-81.

[42] Kingma D P, Ba J. Adam: A Method for Stochastic Optimization[J]. Computer Science, 2014.

[43] Chen C, Ng V. SinoCoreferencer: An End-to-End Chinese Event Coreference Resolver[C]//LREC. 2014: 4532-4538.