SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

November 2019, Vol. 62 212101:1–212101:9
https://doi.org/10.1007/s11432-018-9833-1

Event co-reference resolution via a multi-loss neural network without using argument information

Xinyu ZUO[1,2], Yubo CHEN[1*], Kang LIU[1,2*] & Jun ZHAO[1,2]

[1]*Institute of Automation, Chinese Academy of Sciences, Beijing* 100190*, China;*
[2]*University of Chinese Academy of Sciences, Beijing* 100049*, China*

Received 16 November 2018/Revised 26 February 2019/Accepted 15 March 2019/Published online 9 October 2019

Abstract Event co-reference resolution is an important task in natural language processing, and nearly all the existing approaches for this task rely on event argument information. However, these methods tend to suffer from error propagation from event argument extraction. Additionally, not every event mention contains all arguments of an event, and the argument information may confuse the model where events contain arguments to detect an event co-reference in real text. Furthermore, the context information of an event is useful to infer the co-reference between events. Thus, to reduce the errors propagated from event argument extraction and use context information effectively, we propose a multi-loss neural network model that does not require any argument information relating to the within-document event co-reference resolution task; furthermore, it achieves a significantly better performance than the state-of-the-art methods.

Keywords event co-reference resolution, neural network, information extraction, multi-loss function, event extraction

Citation Zuo X Y, Chen Y B, Liu K, et al. Event co-reference resolution via a multi-loss neural network without using argument information. Sci China Inf Sci, 2019, 62(11): 212101, https://doi.org/10.1007/s11432-018-9833-1

1 Introduction

Event co-reference resolution (ECR) is a task to determine the event mentions in a document that refer to the same real-world event. Event co-reference resolution is a critical part of natural language processing (NLP) systems such as summarization [1], text-level event extraction [2], and question answering [3]. Additionally, compared to considerable research in entity co-reference resolution, less attention is given to event co-reference resolution. Therefore, event co-reference resolution is still a challenging task and its performance should be improved.

Event mentions that refer to the same event can occur both within a document (WD) and across multiple documents (CD). We focus on WD event co-reference herein because it is the basic component of CD event co-reference. The primary task of WD event co-reference is to assess whether a pair of events is co-referential. Figure 1 shows two co-referential event pairs from two documents. The first event pair in D1 is regarding a shooting event and the second event pair in D2 is regarding a fire event.

To assess the co-reference of an event pair, most approaches for solving event co-reference resolution rely on various linguistic properties, especially event argument, which contains spatio-temporal information of events [4]. For instance, in Figure 1, the words in red font are events. Meanwhile, the words in blue, green, and orange fonts indicate the participant, time, and location of the events, respectively.

* Corresponding author (email: yubo.chen@nlpr.ia.ac.cn, kliu@nlpr.ia.ac.cn)

D1:	S1:	Suspended [worker]$_{\text{part}}$ kills [2 women]$_{\text{part}}$ in [kraft]$_{\text{loc}}$ factory [shooting]$_{\text{event1}}$ happened shortly after [8:30 p.m.]$_{\text{time}}$.
	S2:	According to police, in the [Friday evening]$_{\text{time}}$, [the women]$_{\text{part}}$ were [shot]$_{\text{event2}}$ on the third floor in the [building]$_{\text{loc}}$.
D2:	S1:	It was probably "chance" when the [Wasilla Bible Church]$_{\text{loc}}$ was [burned]$_{\text{event1}}$, and arson was suspended.
	S2:	When the [fire]$_{\text{evnet2}}$ destroyed everything, I suspected foul play, but knew that it was probably just chance.

Figure 1 (Color online) Instances of event co-reference resolution.

Although event arguments contain useful information for the event co-reference resolution, two problems exist when using the event argument information in the event co-reference resolution. First, it is difficult to extract event arguments accurately owing to the diverse event argument expressions. The performance of the event argument extraction is only 55.7% [5] in the ACE corpus. For instance, in D1, the arguments regarding a shooting event in the two sentences are the same but expressed differently. In detail, in D1, the participant, time, and location of a shooting event are worker, 2 women, 8:30 p.m., and Kraft in S1, respectively, but women, Friday evening, and building in S2, respectively. Next, not every event mention contains all arguments of one event and may confuse the model regarding the co-reference of two events in an event pair. For instance, in D2, the Wasilla Bible Church for the location of a fire event is in S1 but not in S2. Additionally, in D2, devoid of event arguments, a burned event and fire event are co-referential in context.

As aforementioned, the arguments of events are difficult to extract. It is also difficult to use arguments to solve all problems of event co-reference resolution even if they are extracted. Thus, information regarding event mentions is more important and effective for the event co-reference resolution. To use the context information efficiently, we propose a multi-loss neural network model (MLNN) that does not need any argument information to accomplish a WD event co-reference resolution task. We propose two submodels that use context information to detect the co-reference of two events in an event pair and train them jointly. One is a classifier that predicts whether the two events in one pair are co-referential, and the other is a scorer that calculates the similarity scores between them to assist in inferring a co-reference.

The final stage of the event co-reference resolution is event clustering. After all event pairs are predicted and scored, we filter the event pairs according to the results of the classifier and scorer. Subsequently, we use a dynamic connectivity algorithm to construct a graph for event clustering. Each node in the graph is an event mention, and each edge between two nodes represents whether the two events are co-referential. Finally, all events connected in one graph are considered to be in one event cluster (event chain).

We evaluate our model on the ECB+ corpus [6] and use B^3 [7], CEAF$_e$ [8], MUC [9], and CoNLL F_1 [10] as measures. The experimental results indicate that our model achieved a significant improvement compared to the state-of-the-art methods that use event argument features.

2 Task description

We adopted the ECB+ corpus, which extends the widely used corpus for the event co-reference resolution task, EventCorefBank (ECB) [4]. An event is when a situation occurs, either in the real world or not [11]. In the ECB+ corpus, an event consists of four components: (1) action: what occurs in the event; (2) participants: who or what is involved; (3) time: when the event occurs; and (4) location: where the event occurs. Each document consists of a set of mentions describing event actions, participants, times, and locations. These mentions relate to different events in a document. Table 1 shows the shooting event components of the sentence, "Suspended worker kills two women in Kraft factory shooting that occurred shortly after 8:30 p.m.." in D1 shown in Figure 1.

To consist with the ECB+ corpus, we use the term event mention herein, a verb or noun phrase that describe events most clearly to refer to the mention of an event action, and event argument to refer to all mentions of the participant, time, and location included in the event. Additionally, we define that two

Table 1 Mentions of event components in ECB+ corpus

Action	Participant	Time	Location
Shooting	Worker/2 women	8:30 p.m.	Kraft

events are co-referential when they refer to the same actual event in the real world. As shown in Figure 1, although we can infer the co-reference easily if we use all event arguments information of an event, not all event arguments are presented in an event. Thus, we must reduce the errors propagated from this issue and event argument extraction, and utilize the context regarding events, which is the most reliable information for inferring a co-reference.

3 Related work

Co-reference resolution, in general, is a difficult NLP task and typically requires sophisticated inferentially based knowledge-intensive models [12]. Extensive studies have focused on the problem of entity co-reference resolution and many techniques have been developed, including rule-based deterministic models [13] that traverse mentions in certain orderings and make deterministic co-reference decisions based on all available information at the time, and supervised learning-based models [14] that utilize rich linguistic features and the annotated corpora to learn more powerful functions.

Event co-reference resolution is a more complex task than entity co-reference resolution [2]; thus, it has been studied less, relatively. Different approaches have been proposed to detect WD co-reference chains. Works specific to WD event co-reference include pairwise classifiers [15–17], the graph-based clustering method [18], information propagation [19], the Markov logic network [20], linguistic features based on the unsupervised method [4], the hierarchical distance-dependent Bayesian model [21], and iterative unfolding interdependency models [22].

Similar to these studies, almost all well-performing methods rely on rich features. These methods require complex and time-consuming feature engineering and result in more error propagations.

4 Methodology

The event co-reference resolution task in this study can be divided into a primary subtask and two secondary subtasks. (1) (primary subtask) event co-reference detection: detecting whether each candidate event pair is co-referential; (2) (secondary subtask) event mention extraction: extracting event mentions; and (3) (secondary subtask) event clustering: grouping event mentions into clusters according to their co-reference.

4.1 Event mention extraction

Previous methods regarding event co-reference resolution rely on rich features based on semi-Markov CRFs [21] to identify event mentions. The features include word-level features such as unigrams, bigrams, POS tags, WordNet hypernyms, synonyms, and FrameNet semantic roles, and phrase-level features such as phrasal syntax(e.g., NP, VP) and phrasal embeddings (constructed by averaging word embeddings). Based on head word matching[1)], 95% of event mentions can be identified in the development set.

To consist with the event co-reference detection model and use less features, we constructed a multilayer feedforward neural network with a cross-entropy objective function to identify whether a candidate word is an event mention. Additionally, our neural network uses only the candidate word, context in a window around the candidate word, POS tags in a window around the candidate word, and the lemma of the candidate word as features. Our model can identify 92% of the event mentions in the same development set as that of Yang et al. [21], which is slightly lower than semi-Markov CRFs.

1) For multiword event mentions, to match with previous methods, we used only the first word and its word embedding to represent event mentions.

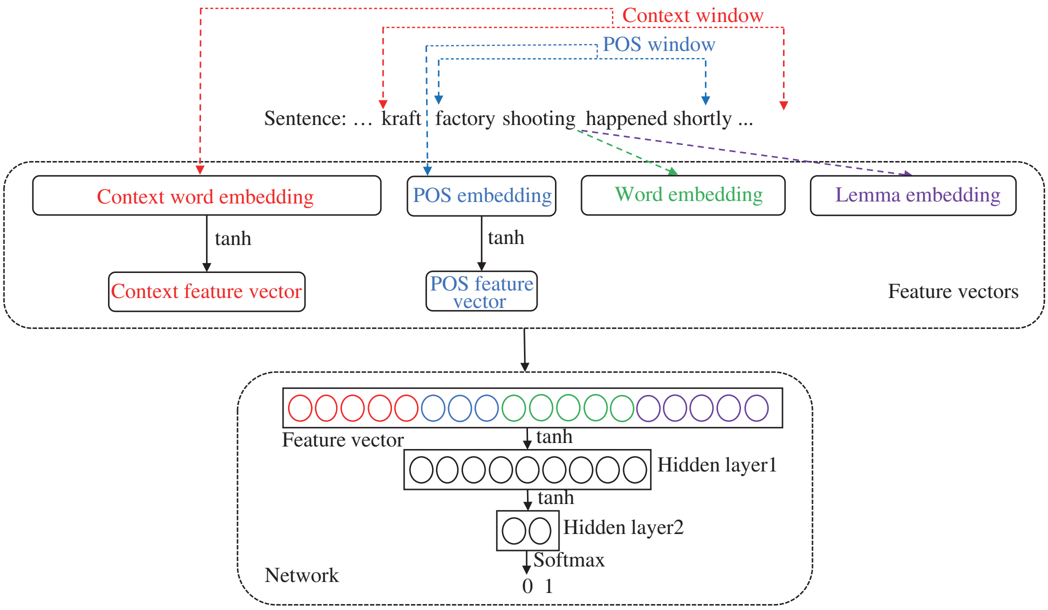

Figure 2 (Color online) Structure of feedforward neural network for event mention extraction.

As shown in Figure 2, we regard the event mention extraction task as a classification task. For each candidate word, we first use the aforementioned features as the input and convert them into context embedding (a combination of words in the window), POS embedding (a combination of POS tags in the window), word embedding, and lemma embedding, separately. Next, we map the context word embedding and POS embedding into a context feature vector and POS feature vector, respectively, by a one-layer feedforward NN. Subsequently, we combine all feature embeddings and pass them through a two-layer feedforward NN that uses tanh as an activation function. Finally, the model outputs a two-dimensional vector in which the value of each dimension is either 0 or 1 after the Softmax operation. The candidate word will be predicted as an event mention if the value of the first dimension is one, and as a non-event mention if the value of the second dimension is one. The objective function of the multilayer feedforward neural network of the event mention extraction is as follows:

$$\mathcal{L}_1(\theta_1) = -\frac{1}{n}\sum_{i=0}^{n}[y_i^e \log \widehat{y}_i^e + (1-y_i^e)\log(1-\widehat{y}_i^e)], \quad y_i^e, \widehat{y}_i^e \in \{0,1\}, \tag{1}$$

$$\widehat{y}_i^e = P(y_i^e = 1|x_i^e), \quad 1-\widehat{y}_i^e = P(y_i^e = 0|x_i^e), \tag{2}$$

where x_i^e is an input event, y_i^e and $\widehat{y}_i^e$ are the correct and predicted labels indicating whether x_i^e is an event or not, respectively. Additionally, n is the quantity of the input event and θ_1 are the parameters.

4.2 Event co-reference detection

We constructed an MLNN model that does not require event argument information and trains the classifier and scorer jointly. The network inputs are a candidate event pair and its features. Further, the system outputs a classification result that indicates whether an event pair is co-referential preliminarily, and a confidence score and similarity score to assist us in inferring the co-reference.

4.2.1 *Event features*

We used features similar to those used in the event mention extraction. (1) context feature: the context around the candidate event in a window; (2) POS feature: the POS tags of words around the candidate event in a window; and (3) lexical feature: the word and its lemma of the candidate event. Similar to event mention extraction, we converted the context feature and POS feature into context word embedding and POS embedding respectively, and subsequently mapped them into a context feature vector and POS

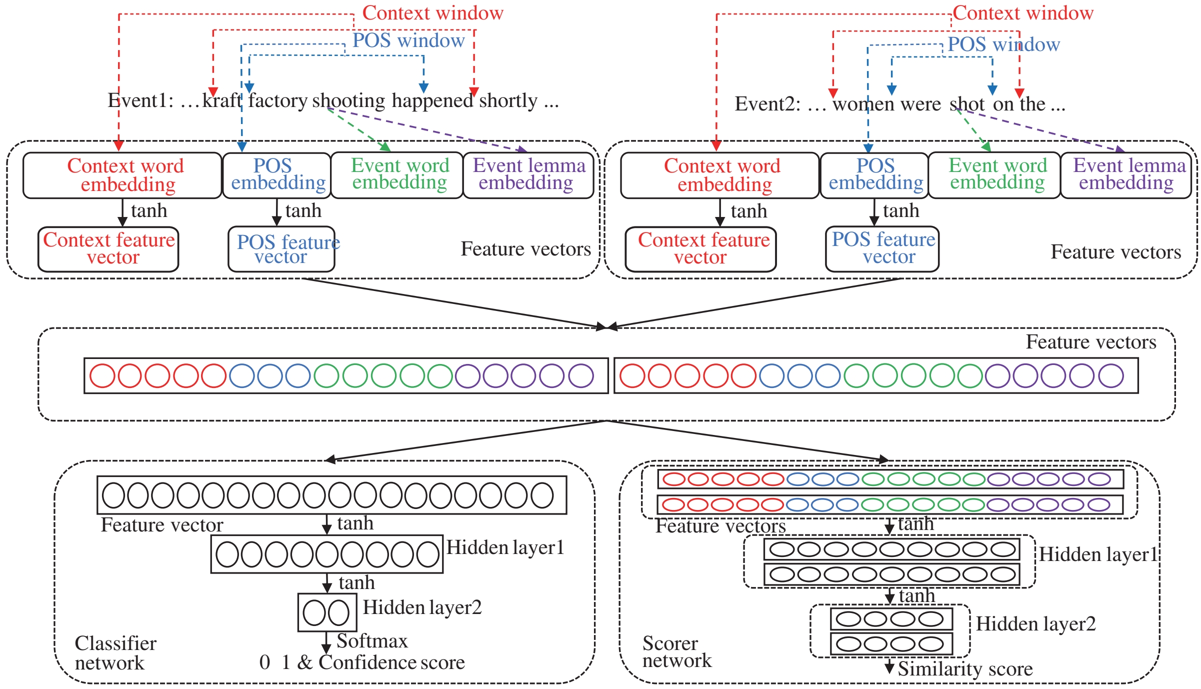

Figure 3 (Color online) Structure of MLNN for event co-reference detection.

feature vector, respectively by a one-layer feed-forward NN. Additionally, we converted the lexical feature into an event word embedding and an event lemma embedding. Finally, we combined the context feature vector, POS feature vector, event word embedding, and event lemma embedding as a feature vector of the candidate event.

4.2.2 *MLNN*

Figure 3 shows the structure of the MLNN for event co-reference detection.

Classification model with cross-entropy objective function. The first subnetwork is a classifier network (CN), as shown in Figure 1. First, we combine the feature vectors of two events in a candidate event pair as one feature vector by stitching and inputting them into the CN. Next, we pass the combined feature vector through a two-layer feedforward NN that uses tanh as an activation function. Finally, after a softmax operation, we obtain the classification results that indicate whether the two events in a candidate event pair is co-referential. Further, the confidence score assists us in inferring the co-reference. Moreover, the cross-entropy objective function of the CN is as follows:

$$\mathcal{L}_2(\theta_2) = -\frac{1}{n}\sum_{i=0}^{n}[y_i \log \widehat{y}_i + (1-y_i)\log(1-\widehat{y}_i)], \quad y_i, \widehat{y}_i \in \{0,1\}, \tag{3}$$

$$\widehat{y}_i = P(y_i = 1|x_i), \quad 1-\widehat{y}_i = P(y_i = 0|x_i), \tag{4}$$

where x_i is the input candidate event pair; y_i and $\widehat{y}_i$ are the correct and predicted labels indicating the co-reference of two events in a candidate event pair, respectively. It indicates a co-referential event pair if the value is zero and not co-referential if the value is one. Additionally, n is the quantity of input event pairs and θ_2 is the parameter of the CN.

Scoring model with similarity difference objective function. The second subnetwork is the scorer network (SN), as shown in Figure 3, which is different from the CN. First, we input the feature vector of two events in a candidate event pair to the SN individually rather than combining them. Next, we pass the feature vectors of two events through a two-layer feedforward NN that uses tanh as an activation function. Finally, we calculate the cosine similarity score of two vectors that is the output of the two-layer NN. Moreover, the similarity difference objective function of the SN is the following:

$$\mathcal{L}_3(\theta_3) = \sum_{i=0}^{n} \log |m_i - s_i|, \quad \begin{cases} m_i = 1, & \text{if } y_i = 0, \\ m_i = -1, & \text{if } y_i = 1, \end{cases} \tag{5}$$

where s_i is the cosine similarity score of the two events in the input event pair. The closer is s_i to 1, the more similar are the two events; the closer it is to -1, the less similar are the two events. Additionally, m_i is the margin of the cosine similarity score.

Joint training with multi-loss function. We train the two submodels jointly and combine their objective functions as the following:

$$\mathcal{L}_{\text{all}}(\theta_{\text{all}}) = \mathcal{L}_2(\theta_2) + \mathcal{L}_3(\theta_3), \tag{6}$$

where θ_{all} are parameters of the whole system.

Therefore, we optimize the cross-entropy and similarity difference objective function simultaneously through $L_{\text{all}}(\theta_{\text{all}})$. In detail, the parameters of the classification model and scoring model can be updated simultaneously by the joint model.

4.3 Event clustering method

After classification and scoring, we filter the event pairs according to the result obtained from the classifier and scorer. Subsequently, we construct a graph to cluster the filtered events by a dynamic connectivity algorithm. Each node in the graph is an event mention and the two events are co-referential if an edge exists between them. The details are as follows.

4.3.1 *Event pair filtering*

The annotations in the ECB+ regarding the event mention and event co-reference are incomplete because some co-references between the events in the real world are not marked in text [6]. This phenomenon propagates an error in which a word is annotated as an event mention when in fact it is not. Thus, we used confidence and similarity scores to enhance the recall regarding event co-reference detection. Eventually, for each input candidate event pair, the two events are co-referential if the classification result is a co-reference. Further, we identified the two events as co-referential if the classification result is a non-co-reference but the similarity score is greater than 0.5 and the confidence score is less than 0.6. The thresholds of the similarity score and confidence score are determined by the development set.

4.3.2 *Event clustering*

We used a dynamic connectivity algorithm to merge two events that we identified as co-referential in each event pair from one document. After merging, we regarded each individual subgraph as a cluster, and events in the same cluster as a co-referential event chain.

5 Evaluation

We performed all experiments on the ECB+ corpus. We adopted the settings of the datasets used in Choubey et al. [22]. We divided the dataset into a training set (topics 1–20), development set (topics 21–23), and test set (topics 34–43), similar to the work of Choubey et al [22]. Table 2 shows the distribution of the corpus.

We evaluated our system with four widely used co-reference resolution metrics: B^3 measures the proportion of overlap between the predicted and gold clusters for each mention, CEAF$_e$ measures the best alignment of the gold-standard and predicted clusters, MUC measures the number of gold cluster merging operations required to recover each predicted cluster, and CoNLL F_1, the most important metric, is the average of the F_1 scores for all the three metrics.

Table 2 Statistics of ECB+ corpus

	Train	Dev.	Test	Total
#Document	462	73	447	982
#Sentences	7294	649	7867	15810
#Event mentions	3555	441	3290	7268
#WD chains	2499	316	2137	4953
Average WD chain length	2.8	2.6	2.6	2.7

We used the Natural Language ToolKit [23] tools to extract the POS and lemma. In detail, we set the window sizes of the context and POS as five and three, respectively. Moreover, we set the sizes of word embedding, POS embedding, lemma embedding as 100, 10, and 100, respectively. Additionally, we minimized the objective function over shuffled mini-batches with the Adadelta [24] update rule and used the publicly available official implementation of the revised co-reference scorer (v8.0.1)[2)].

5.1 Baseline and our systems

We compare our model with five baselines[3)].

LEMMA. The first baseline event mentions are grouped into clusters if they have the same lemmatized head word. This is considered as a strong baseline.

HDDCRP [21]. The second baseline is the supervised hierarchical distance-dependent Bayesian model on the ECB+ corpus. This model utilizes the distances between event mentions generated using a feature-rich learnable distance function, as Bayesian priors for single pass nonparametric clustering.

HDP-LEX [4]. The third baseline is an unsupervised hierarchical Bayesian model by Bejan et al. [4].

Agglomerative [16]. The fourth baseline is a two-step agglomerative clustering model.

Iterative WD/CD classifier (Iter-WD/CD) [22]. This baseline is an iterative event co-reference model by exploiting the interdependencies of both WD and CD event mentions.

All five baselines used the event argument features, and the five models were tested on the same dataset as the above.

Our system. First, we detected and filtered the event mentions annotated in the corpus by our event mention extraction model before event co-reference detection and clustering, because the annotations regarding event mentions and co-reference were incomplete. Next, to demonstrate the performance of our MLNN, we evaluated three different systems in terms of event co-reference detection and clustering. (1) C-NN. We only trained the classification model with the cross-entropy objective function and used the results of the classifier to infer the co-reference. (2) C-MLNN. We trained the full MLNN model, but used only the classifier results to infer the co-reference. (3) MLNN. We used a full MLNN model with the classifier and scorer (co-reference classification results as well as the non-co-reference classification result with the confidence and similarity scores).

5.2 Results

Table 3 shows the results of the WD document event co-reference resolution on the ECB+ corpus. In Table 3, the results in bold are the best performance of each metric, and the results in Italics are the second best performance of each metric. We found that although we did not use any features regarding event argument, which are crucial information when understanding events, and the CRF method used in other systems performed slightly little better than ours in event mention extraction, the performance of CoNLL F_1 in our MLNN model is noticeably better than the state-of-the-art methods. In detail, although the F_1 value in CEAF$_e$ is slightly lower, we obtained the second best F_1 value in MUC and the best F_1 value in B^3. In general, an obvious improvement is observed regarding the performance of the WD event co-reference resolution using the MLNN model.

2) https://github.com/conll/reference-coreference-scorers.

3) The results are extracted from the Coubey et al..

Table 3 Results of within-document event co-reference resolution on ECB+ corpus

	B^3			MUC			CEAFE$_e$			CoNLL F_1
	R	P	F_1	R	P	F_1	R	P	F_1	F_1
LEMMA	56.8	*80.9*	66.7	35.9	**76.2**	48.8	67.4	62.9	65.1	60.2
HDP-LEX (2010)	67.6	74.7	71.0	39.1	50.0	43.9	71.4	66.2	68.7	61.2
Agglomerative (2009)	67.6	80.7	73.5	39.2	61.9	48.0	*76.0*	65.6	70.4	63.9
HDDCRP (2015)	67.3	**85.6**	*75.4*	41.7	*74.3*	53.4	**79.8**	65.1	*71.7*	66.8
Iter-WD/CD (2017)	*69.2*	76.0	72.4	*58.5*	67.3	**62.6**	67.9	**76.1**	**71.8**	*68.9*
MLNN	**87.3**	71.0	**78.3**	**69.0**	57.0	*62.4*	66.6	*76.0*	70.7	**70.4**

Table 4 Comparisons of three systems

	B^3			MUC			CEAFE$_e$			CoNLL F_1
	R	P	F_1	R	P	F_1	R	P	F_1	F_1
C-NN	**90.2**	48.8	63.3	**76.8**	40.0	56.0	40.2	69.7	51.0	56.8
C-MLNN	86.8	67.7	76.0	67.6	53.3	59.6	62.3	74.5	67.9	67.8
MLNN	87.3	**71.0**	**78.3**	69.0	**57.0**	**62.4**	**66.6**	**76.0**	**70.7**	**70.4**

In Table 4, first, the results of the C-NN and C-MLNN indicate that classification model training with the scoring model can improve the performance of the classifier to infer an event co-reference. Next, the results of the C-MLNN and MLNN indicate that the scores from the scorer can assist the classifier in improving the performance of the WD event co-reference resolution.

6 Error analysis

Two primary issues cause errors in this study.

6.1 Incorrect co-reference links

As our method utilizes contextual information primarily, events in adjacent positions in the same sentence but from different event chains may be inferred as co-referential events. For instance, in the sentence, "This anti-piracy action by INS Sukanya was the fifth successful operation by its crew during its current patrol mission in the Gulf of Aden since September this year," the two events, i.e., patrol event and mission event are not co-referential but have similar context.

6.2 Incomplete annotation

As mentioned above, the annotations of event mention and co-reference is incomplete; for example, in the sentence, "INS Sukanya has sized 14 AK-47 rifles, 31 magazines, and 923 rounds of ammunition during the five operations it that it performed, the officer said...," sized is an event mention but not marked in the ECB+ corpus. The contextual information of the event in the text is vital for event mention extraction and event co-reference detection. However, the context of the incomplete annotated events will result in confusion to the system for the two subtasks in the training stage.

7 Conclusion and future work

We presented a multilayer feedforward neural network for event mention extraction and an MLNN model for WD event co-reference resolution. We did not use any information regarding event arguments in our system. Additionally, we tested our system with the ECB+ corpus and achieved a significant improvement over the state-of-the-art methods.

Owing to the incomplete annotation and the propagation of errors regarding event mentions and argument extractions in pipeline systems, we will attempt to design a joint model to accomplish event extraction, argument extraction, and event co-reference resolution tasks jointly in the future.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61533018, 61806201, 61702512), Independent Research Project of National Laboratory of Pattern Recognition. This work was also supported by CCF-Tencent Open Fund.

References

1 Daniel N, Radev D, Allison T. Sub-event based multi-document summarization. In: Proceedings of HLT-NAACL 03 on Text Summarization Workshop, 2003. 9–16
2 Humphreys K, Gaizauskas R, Azzam S. Event coreference for information extraction. In: Proceedings of Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts, 1997. 75–81
3 Narayanan S, Harabagiu S. Question answering based on semantic structures. In: Proceedings of the 20th International Conference on Computational Linguistics, 2004
4 Bejan C, Harabagiu S. Unsupervised event coreference resolution with rich linguistic features. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 2010. 1412–1422
5 Chen Y B, Liu S L, Zhang X, et al. Automatically labeled data generation for large scale event extraction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017. 409–419
6 Cybulska A, Vossen P. Guidelines for ECB+ Annotation of Events and Their Coreference. Technical Report NWR-2014-1. 2014
7 Bagga A, Baldwin B. Algorithms for scoring coreference chains. In: Proceedings of the 1st International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference, 1998. 563–566
8 Luo X Q. On coreference resolution performance metrics. In: Proceedings of Conference on Human Language Technology and Empirical Methods in Natural Language Processing, 2005. 25–32
9 Vilain M, Burger J, Aberdeen J, et al. A model-theoretic coreference scoring scheme. In: Proceedings of the 6th Conference on Message Understanding, 1995. 45–52
10 Pradhan S, Luo X Q, Recasens M, et al. Scoring coreference partitions of predicted mentions: a reference implementation. In: Proceedings of Conference Association for Computational Linguistics Meeting, 2014
11 Cybulska A, Vossen P. Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution. In: Proceedings of Language Resources and Evaluation, 2014. 4545–4552
12 Kehler A, Kehler A. Coherence, Reference, and the Theory of Grammar. Stanford: CSLI Publications, 2002
13 Cardie C, Wagstaf K. Noun phrase coreference as clustering. In: Proceedings of Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999
14 Stoyanov V, Gilbert N, Cardie C, et al. Conundrums in noun phrase coreference resolution: making sense of the state-of-the-art. In: Proceedings of Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2009. 656–664
15 Ahn D. The stages of event extraction. In: Proceedings of Workshop on Annotating and Reasoning About Time and Events, 2006
16 Chen Z, Ji H, Haralick R. A pairwise event coreference model, feature impact and evaluation for event coreference resolution. In: Proceedings of Workshop on Events in Emerging Text Types, 2009. 17–22
17 Zeng D J, Dai Y, Li F, et al. Adversarial learning for distant supervised relation extraction. Comput Mater Con, 2018, 55: 121–136
18 Chen Z, Ji H. Graph-based event coreference resolution. In: Proceedings of Workshop on Graph-based Methods for Natural Language Processing, 2009. 54–57
19 Liu Z Z, Araki J, Hovy E H, et al. Supervised within-document event coreference using information propagation. In: Proceedings of Language Resources and Evaluation, 2014. 4539–4544
20 Lu J, Venugopal D, Gogate V, et al. Joint inference for event coreference resolution. In: Proceedings of the 26th International Conference on Computational Linguistics, 2016. 3264–3275
21 Yang B S, Cardie C, Frazier P. A hierarchical distance-dependent Bayesian model for event coreference resolution. Trans Assoc Comput Linguist, 2015, 3: 517–528
22 Choubey P K, Huang R. Event coreference resolution by iteratively unfolding inter-dependencies among events. 2017. ArXiv:1707.07344
23 Bird S, Klein E, Loper E. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. Sebastopol: O'Reilly Media Inc., 2009
24 Zeiler M D. ADADELTA: an adaptive learning rate method. 2012. ArXiv:1212.5701