Employing Multiple Decomposable Attention Networks to Resolve Event Coreference

Jie Fang, Peifeng Li(✉), and Guodong Zhou

School of Computer Science and Technology,
Soochow University, Suzhou, China
jfang@stu.suda.edu.cn, {pfli,gdzhou}@suda.edu.cn

Abstract. Event coreference resolution is a challenging NLP task due to this task needs to understand the semantics of events. Different with most previous studies used probability-based or graph-based models, this paper introduces a novel neural network, MDAN (Multiple Decomposable Attention Networks), to resolve document-level event coreference from different views, i.e., event mention, event arguments and trigger context. Moreover, it applies a document-level global inference mechanism to further resolve the coreference chains. The experimental results on two popular datasets ACE and TAC-KBP illustrate that our model outperforms the two state-of-the-art baselines.

Keywords: Event coreference · Decomposable Attention Network
Global inference

1 Introduction

Event coreference resolution is vital for many NLP applications, such as topic detection (Allan et al. [1]), information Extraction (Li et al. [2]) and question answering (Narayanan and Harabagiu [3]). It is to determine which event mentions in texts refer to the same real-world event and then cluster them to a unique coreferential event chain. Take the following two event mentions as samples:

S1: A Cuban patrol boat with four heavily armed men **landed** on American shores.
S2: These bozos let four armed Cubans **land** on our shores.

The event mention in S1, whose event trigger is "landed", and the mention in S2 with trigger "land" refer to the same real-world Movement event, and are coreferential event mentions.

This paper focuses on document-level event coreference resolution. Document-level event coreference chains are challenging to resolve. Sometimes, coreferential event mentions in the same document can look very dissimilar ("killed/VB" and "murder/NN"), have event arguments partially or entirely omitted, or appear in distinct contexts compared to their antecedent event mentions, partially to avoid repetitions.

To capture the semantic information hiding in event trigger, event argument and the structure among the trigger and its arguments, this paper introduces a novel neural network, MDAN (Multiple Decomposable Attention Network) (Parikh et al. [4]), to resolve document-level event coreference from different views, i.e., event mention,

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11109, pp. 246–256, 2018.
https://doi.org/10.1007/978-3-319-99501-4_21

event arguments and trigger context. This model can capture the different features from different views to better represent the event semantics. To resolve conflicts between different event mention pairs, this paper applies a document-level global inference mechanism to further resolve the coreference chains. The experimental results illustrate that our model outperforms the two state-of-the-art baselines on two popular datasets, the ACE 2005 corpus and the TAC KBP 2015 corpus. The contributions of this paper are as follows:

It constructs a novel neural network model MDAN for document-level event coreference resolution to capture different event semantics from multiple views.

It introduces event arguments to MDAN to better represent event semantics.

It applies a document-level global inference mechanism to further resolve the coreference chains.

The rest of this paper is organized as follows. Section 2 overviews the related work. Section 3 describes our MDAN model for event coreference resolution. Section 4 evaluates our approach and shows its effectiveness over two baselines. Section 5 concludes the paper with future work.

2 Related Work

Event coreference is much less studied in comparison to the large number of work on entity coreference. The studies on event coreference resolution are usually divided into within-document level and cross-document level.

Early work on document-level event coreference resolution mostly built on insights gained from the entity coreference literature (Cybulska and Vossen [5], Bejan and Harabagiu [6], Ng and Cardie [7]). Recent approaches focused on exploiting event specific structure and resolution model. Chen and Ji [8] modeled event coreference resolution as a spectral graph clustering problem that optimizes the normalized-cut criterion. Liu et al. [9] introduced a rich-features method with a large amount of features for propagating information between events and their arguments. Lu et al. [10] proposed a joint inference model based Markov logic networks to correct the mistakes from the pairwise event coreference resolver. Currently, neural networks are widely used in many NLP applications. To our knowledge, there is only one study employed neural networks for document-level event coreference. Krause et al. [11] introduced the Convolutional Neural Network (CNN) to event coreference. It is divided into two parts. The first part gives a representation for a single event mention and the second part is fed with two such event mention representations plus a number of pairwise features for the input event-mention pair, and calculates a coreference score.

3 MDAN for Event Coreference Resolution

The architecture of the model MDAN for event coreference resolution is shown in Fig. 1 and our model MDAN contains four parts, i.e., multi-similarity module, pairwise module, classifier module and global inference module.

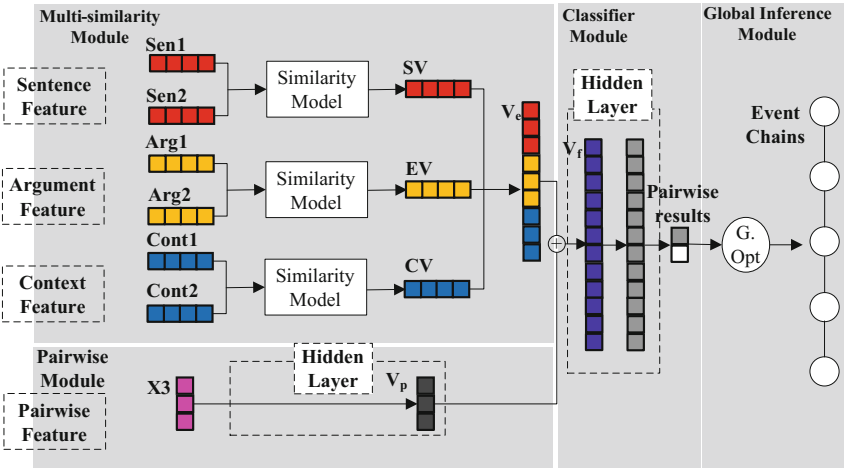

Fig. 1. The architecture of the MDAN model

Two event mentions are coreferential when they are similar in tokens, event structures, triggers, arguments, context of trigger, etc. The multi-similarity module first compute the similarity vectors of two event mentions from multiple views (i.e., event mention view, argument view and trigger context view), and then concatenate them into a vector to represent their final similarity. The advantage of multi-similarity module is that it can capture different semantic information from different views to represent different similarities of an event pair. The details of a single similarity module are shown in Fig. 2. Inspired by Parikh et al. [4], we introduce Decomposable Attention Network (DAN) as our similarity module. DAN outperforms several similarity models in our experiments, such as Siamese CNN Network, etc. Its advantage is that the soft attention in DAN can capture important hiding features and avoid noise. This similarity module containing the soft attention is suitable to learn the similarity of two event mentions in our experiments.

The pairwise module fed with pairwise features between two event mentions and maps them to a vector. Pairwise features reveal the similarities on various kind attributes (e.g., trigger and event type) of event mentions. These attributes can be regarded as the auxiliary of the multi-similarity module. Both the multi-similarity module and the pairwise module are the kernel components of MDAN, and we use them to capture the hidden features inside event mention.

The classifier module is to classify an event mention pair to coreference or not. The input of this module is the combination of the event similarity vector from the multi-similarity module and the pairwise vector from the pairwise module.

The global inference module is to optimize the results from the classifier module to form a more complete event chain, based on the merging and cutting rules.

In our model MDAN, the inputs of the multi-similarity module and the pairwise module are the extracted features from an event mention pair, while the output of the classifier module is the confidence score that two event mentions are coreferential. The

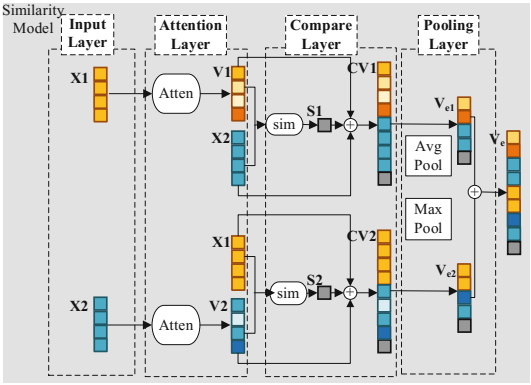

Fig. 2. The architecture of the DAN-based similarity model

input of the global inference module are all the confidence scores of two event mentions in a document and its output is the optimized results of event chains.

3.1 Input

Following Krause et al. [11], the input of the model MDAN is two event mentions e1 and e2 with annotated trigger, event type/subtype, event arguments, and event attributes (e.g., modality), etc. We extract the features from these two event mentions and the features used in Krause et al. [11] are employed in our model as follows.

Event features: (Multi-similarity module)

- Sentential features: words in sentences (event mentions) (F1); relative positions of words based on triggers (e.g., that of the word "shores" in S1 is 3) (F2)
- Context features: context around trigger (the windows size is set to 5, e.g., "with four heavily armed men landed on American shores" in S1) (F3).

Pairwise features: (Pairwise module)

- Event type and subtype is the same or not (F4)
- Distance between event mentions (numeric values) (F5)
- Event modality is the same or not (F6)
- Overlap in arguments or not (F7).

To further capture the semantic information in sentence structures and arguments (Haghighi and Klein [12]), we provide the additional features as follows:

Event features: (Multi-similarity module)

- Sentential features: POS of words in sentences (event mentions) tagged by NLTK tools (F8)
- Argument features: arguments in sentences (F9) and their entity types (F10).

For each event mention, we first embed the features (F1–F3 and F8–F10) in the sets of the sentential features, context features and argument features to six vectors, where

the features F1, F3 and F9 are embedded by word2vec and the others are embedded randomly. Then we concatenate the vectors from F1, F2 and F8 into the vector **Sen**, the vectors from F9 and F10 into the vector **Arg**. Besides, the vector **Cont** is the vector from F3. In Fig. 1, **Sen1**, **Cont1** and **Arg1** are the vectors of the sentential features, context features and argument features for event mention e1, while **Sen2**, **Cont2** and **Arg2** are the vectors for event mention e_2.

3.2 Multi-similarity Module

Multi-similarity module contains three DAN-based similarity models showed in Fig. 2. Each DAN-based similarity model compute the similarity between two event mentions e_1 and e_2 on three different views, i.e., sentence, context and argument views, by using pairwise vector ***Sen1-Sen2***, ***Cont1-Con2*** and ***Arg1-Arg2*** as input, respectively.

In each similarity model, we first employ the attention mechanism to calculate the weights of input vectors ***X1*** and ***X2*** (i.e., ***Sen1-Sen2***, or ***Cont1-Con2***, or ***Arg1-Arg2***), for extracting important information from two given event mentions. Our soft alignment layer computes the attention weights w_{ij} as the similarity of words in the tuple ***<X1, X2>*** as Eq. (1) where function ***F*** is a feed-forward neural network, and $\boldsymbol{X1}_i$ is the vector of ith word in the vector ***X1***:

$$w_{ij} = F(\boldsymbol{X1}_i)^T \cdot F(\boldsymbol{X2}_j) \tag{1}$$

Then we use softmax to compute the weight of vectors. Vectors ***X1*** and ***X2*** are normalized as ***V1*** and ***V2*** as follows, where ℓ_{X1} and ℓ_{X2} is the length of ***X1*** and ***X2***, $\boldsymbol{V1}_i$ is ith word of ***X*** after adding attention weight values:

$$\begin{aligned} \mathbf{V1}_{\mathrm{i}} &= \sum\nolimits_{\mathrm{j}=1}^{\ell_{\mathbf{X1}}} \frac{\exp(\mathrm{w_{ij}})}{\sum_{\mathrm{k}=1}^{\ell_{\mathbf{X1}}} \exp(\mathrm{w_{ik}})} \mathbf{X1}_{\mathrm{j}} \ \forall \mathrm{i} \in [1, \ldots, \ell_{\mathbf{X1}}] \\ \mathbf{V2}_{\mathrm{i}} &= \sum\nolimits_{\mathrm{j}=1}^{\ell_{\mathbf{X2}}} \frac{\exp(\mathrm{w_{ij}})}{\sum_{\mathrm{k}=1}^{\ell_{\mathbf{X2}}} \exp(\mathrm{w_{ik}})} \mathbf{X2}_{\mathrm{j}} \ \forall \mathrm{i} \in [1, \ldots, \ell_{\mathbf{X2}}] \end{aligned} \tag{2}$$

Equation (3), which can be viewed as a noisy channel, is to compute the cosine distance of two vectors, so we can get the similarity scores of two vectors S1 and S2 as follows:

$$\mathbf{S1} = \mathrm{sim}(\mathbf{X2}, \mathbf{V1}) = \mathbf{X2}^{\mathrm{T}} \cdot \mathbf{V1}, \ \mathbf{S2} = \mathrm{sim}(\mathbf{X1}, \mathbf{V2}) = \mathbf{X1}^{\mathrm{T}} \cdot \mathbf{V2} \tag{3}$$

Then we concatenate *X2*, *V1* and *S1* into a comparison vector *CV1*, and the same for comparison vector *CV2*:

$$\boldsymbol{CV1} = [\boldsymbol{X2}, \boldsymbol{V1}, \boldsymbol{S1}], \ \boldsymbol{CV2} = [\boldsymbol{X1}, \boldsymbol{V2}, \boldsymbol{S2}] \tag{4}$$

Next, we use pooling to reduce the complexity of representation. Maximum Pooling (MaxPool) and Averaging Pooling (AvgPool) are two main approaches of pooling. However, AvgPool may weaken strong activation values, and MaxPool may

lead to overfitting. So we compute both average pooling (i.e., $PV1_{avg}$ and $PV2_{avg}$) and max pooling (i.e., $PV1_{max}$ and $PV2_{max}$), and concatenate all of them to produce the vector V_e:

$$\boldsymbol{PV1_{avg}} = \sum_{i=1}^{\ell_{CV1}} \frac{\boldsymbol{CV1_i}}{\ell_{\boldsymbol{CV1}}}, \quad \boldsymbol{PV1_{max}} = \boldsymbol{max}_{i=1}^{\ell_{CV1}} \boldsymbol{CV1_i} \tag{5}$$

$$\boldsymbol{PV2_{avg}} = \sum_{j=1}^{\ell_{CV2}} \frac{\boldsymbol{CV2_j}}{\ell_{\boldsymbol{CV2}}}, \quad \boldsymbol{PV2_{max}} = \boldsymbol{max}_{j=1}^{\ell_{CV2}} \boldsymbol{CV2_j} \tag{6}$$

$$\mathbf{V}_e = \left[\boldsymbol{PV}1_{avg}, \boldsymbol{PV}1_{max}, \boldsymbol{PV}2_{avg}, \boldsymbol{PV}2_{max}\right] \tag{7}$$

3.3 Pairwise Module

The pairwise module is to judge the similarity between two event mentions on the pairwise features. This model is very simple. We first transfer the numeric values of to the vector *X3*, and then feed them into a feed-forward network to get the vector V_p for extracting features hiding in pairwise features.

3.4 Classifier Module

We first concatenate V_e from the multi-similarity module and V_p from the pairwise model into the vector V_f to represent the final semantic relation between two event mentions.

$$\boldsymbol{V}_f = \left[\boldsymbol{V}_e, \boldsymbol{V}_p\right] \tag{8}$$

The final vector V_f is fed into a final multilayer perceptron (MLP) classifier, which has three hidden layers with RELU activation, as following:

$$\boldsymbol{V}_{\mathrm{h}} = \alpha\left(\boldsymbol{W}_h * \boldsymbol{V}_f + \boldsymbol{b}\right) \tag{9}$$

where α is the activation function, W_h and b are parameters. Finally, we can get the coreference score with the output of sigmoid layer:

$$Score = sigmoid(\boldsymbol{W}_{out} * \boldsymbol{V}_h + \boldsymbol{b}_{out}) \tag{10}$$

The objective function of model is set as following:

$$\mathcal{J}(\theta) = -\log \prod_{i=1}^{N} p\left(y^{(i)} / x^{(i)}, \theta^{(i)}\right) + \frac{\lambda}{2}\|\theta\|^2 \tag{11}$$

where $\theta = \{\boldsymbol{W}_{X1}, \boldsymbol{W}_{X2}, \boldsymbol{W}_{V1}, \boldsymbol{W}_{V2}, \boldsymbol{W}_{CV1}, \boldsymbol{W}_{CV2}, \boldsymbol{W}_e, \boldsymbol{W}_p, \boldsymbol{W}_f, \boldsymbol{W}_h, \boldsymbol{W}_{out}, \boldsymbol{b}_{out}\}$. To prevent overfitting, we utilize dropout and batch normalization.

3.5 Global Inference Module

To ensure consistent outputs on a document, we propose a document-level global inference module to solve the conflicting decisions in MDAN. We transfer the pairwise results to form coreference chains. The coreferential events are characterized as reflexive, symmetric and transitive. We utilize the transitive property in coreferential events following the merging and cutting rules.

Merging: if both event mention pairs (e_i, e_k) and (e_k, e_j) are coreferential, coreferential relation must hold between e_i and e_j, with event mention e_k as a bridge to link e_i and e_j.

Cutting: if event mention pairs (e_i, e_k) are coreferential and (e_k, e_j) are not coreferential, the pair (e_k, e_j) are not coreferential. If this constraint is in conflict with the merging rule, the merging rule is prior to this cutting rule.

To avoid the conflicts of the above four rules, we count the numbers of coreference and not coreference judgements, respectively, and make final decisions with Eq. (12) as follows.

$$\text{argmax}_x\left(x * \text{CR}\left(e_i, e_j\right) + (1 - \text{x}) * \text{CUR}\left(e_i, e_j\right)\right) \tag{12}$$

where x is a binary indicator. If x equals to 1, the event mention pair (e_i, e_j) is coreferential; otherwise, they are not coreferential. CR(e_i, e_j) and CUR(e_i, e_j) are the count of the results Coref(e_i, e_j) and Uncoref(e_i, e_j) infer by above four rules.

4 Experiments

In this section, we first introduce the experimental setting and then evaluate our model MDAN on two corpora to justify its effectiveness and report the experimental results. Finally, we give the analysis on the experimental results.

4.1 Experimental Setting

In our experiments, we mainly evaluate our MDAN model on the ACE 2005 English corpus, following most previous studies on document-level event coreference resolution. This corpus contains 599 documents in six genres. This corpus annotated events with 8 event types and 33 event subtypes. In our evaluation, we use the same training and test set as Krause et al. [11][1]. Every event mention is paired with every event mention in the text. Besides, we also report the results of our MDAN on another widely used corpus, the TAC KBP 2015 English corpus which is annotated with event nuggets that fall into 38 types and coreference relations between event mentions. Table 1 shows the statistics on the above two corpora.

In the evaluation, we set the dimensions of the POS, entity type, and relative position embeddings as 50 and $\lambda = 10^{-4}$, which parameters of embedding matrix are randomly initialized. We initialize word embeddings via pre-trained embeddings of

[1] https://git.io/vwEEP.

Table 1. Statistics on the ACE and TAC KBP corpora.

Corpus	#Documents	#Sentences	#Event mentions	# Event chains
ACE 2005	599	15494	5268	4046
TACKBP 2015	360	15824	12976	7415

GloVe and set the dimensions as $d_0 = 50$. Besides, we employ mini-batch SGD algorithm to optimize our models and the model training is run for 15 epochs, after which the best model on the valid dataset is selected.

We compare all systems using four standard F1 metrics in previous work: a link-level metric MUC, a mention-level metric B3, an entity-level metric CEAFe and an average pairwise-positive and pairwise-negative F1-score metric BLANC. (Vilain et al. [13]) We also use the average scores (AVG) of the above metrics as comparison metric.

4.2 Experimental Results

To evaluate the performance of our MDAN model on document-level event coreference resolution, we compare it with two strong baselines: a state-of-the-art classifier model (Liu et al. [9]) with more than 100 features and a state-of-the-art neural network model (Krause et al. [11]). Table 2 illustrates the performance comparison on three models based on annotated event mentions.

Table 2. Performance of the model MDAN & competitors on ACE corpus.

System	BLANC			B^3			MUC			AVG
	P	R	F1	P	R	F1	P	R	F1	F1
Liu	70.01	70.88	70.43	88.86	89.90	89.38	48.75	53.42	50.98	70.26
Krause	71.80	75.16	73.31	86.12	90.52	88.26	45.16	61.54	52.09	71.22
MDAN	**80.87**	**86.28**	**83.29**	**89.34**	**90.71**	**90.02**	**65.69**	**65.21**	**65.45**	**79.58**

The results in Table 2 show that our model MDAN outperforms two baselines on all three metrics and their averages, with an average gain of 9.32 and 8.36 in F1-scores, respectively. Compared to baseline Krause, our MDAN improves the F1-scores on three metrics BLANC, B^3 and MUC by 9.98, 1.76 and 13.36, respectively. These results confirm that our decomposable attention network and the global inference mechanism are better than their CNN model and rules on transitive closure.

We also evaluate our MDAN model on another popular event coreference corpus, the TAC KBP 2015 English corpus. Table 3 shows the F1-scores of our MDAN and the top system (TAC-TOP) [12] in the 2015 TAC KBP Event Nugget Evaluation Task. Due to this corpus did not annotate argument tags, we use PractNLPTools[2] to extract event arguments automatically.

[2] https://github.com/biplab-iitb/practNLPTools.

Table 3. F1-scores of MDAN and top system (TAC-TOP) on the 2015 TAC-KBP event nugget evaluation task. (Lu and Ng [15])

System	BLANC	B^3	MUC	$CEAF_e$	AVG
TAC-TOP	76.91	82.29	68.08	74.12	75.35
MDAN	**76.97**	**82.36**	**69.88**	**76.65**	**76.46**

Table 3 shows that our model MDAN outperforms TAC-TOP in all metrics and this result further ensures the effectiveness of our MDAN. Compared with the work of TAC-TOP (Mitamura et al. [14]), which used additional semantic resources and additional annotated datasets, we did not use any external resources.

4.3 Analysis

Compared to the baselines, our improvements mainly derive from three aspects: (1) argument information, (2) MDAN model and its attention mechanism, and (3) global inference mechanism. Table 4 shows the performance when we remove the argument information, or the attention mechanism, or the global inference mechanism from MDAN, respectively.

Table 4. Performance of our MDAN without argument information (Arg)/attention mechanism (Att)/global optimization method (Opt).

System	BLANC			B^3			MUC			AVG
	P	R	F1	P	R	F1	P	R	F1	F1
MDAN	**80.87**	**86.28**	**83.29**	**89.34**	**90.71**	**90.02**	**65.69**	**65.21**	**65.45**	79.58
w/o Arg	73.66	77.44	75.36	88.2	88.98	88.59	61.31	60.86	61.09	75.01
w/o Att	64.61	80.51	67.91	73.12	**92.76**	81.78	45.37	**74.63**	56.43	68.7
w/o Opt	68.76	79.5	72.37	77.81	91.61	84.15	49.5	71.73	58.57	71.69

If we remove the argument information from MDAN, Table 4 shows that the F1-scores on the metrics BLANC, B3 and MUC are reduced −7.93, −1.43 and −4.36, respectively. These results prove that argument information is helpful to identify event mentions and their coreference, because event semantics not only derives from trigger semantics, but entity semantics.

Table 4 also shows that the attention mechanism is helpful to prevent the interference from the uncorrelated features and improves the F1-scores on the metrics BLANC, B^3, and MUC significantly. The principle of our attention mechanism is to weight different input information. Compared with MDAN w/o Opt, We found our MDAN's recall decreased, because the vectors with low attention weight values will be ignored.

We also use the Krause's rules to replace our global inference mechanism, and the results are shown in Table 4 (MDAN w/o Opt). The results show that our global inference mechanism outperforms Krause's rules on all metrics. The reason is that our

global inference mechanism applies both merging and cutting rules to optimize the results of the neural network model and then can balance the output results.

5 Conclusion

This paper introduces a novel neural network MDAN to resolve document-level event coreference from different views. Moreover, it applies a document-level global inference mechanism to further resolve the coreference chains. The experimental results illustrate that our model outperforms the two state-of-the-art baselines on two popular datasets ACE and TAC-KBP. Our future work is to expand our model to cross-document and multi-language event coreference resolution.

Acknowledgments. The authors would like to thank three anonymous reviewers for their comments on this paper. This research was supported by the National Natural Science Foundation of China under Grant Nos. 61772354, 61773276 and 61472265, and was also supported by the Strategic Pioneer Research Projects of Defense Science and Technology under Grant No. 17-ZLXDXX-02-06-02-04.

References

1. Allan, J., Carbonell, J.G., Doddington, G., Yamron, J., Yang, Y.: Topic detection and tracking pilot study: final report. In: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, pp. 194–218 (1998)
2. Li, P., Zhu, Q., Zhou, G.: Argument inference from relevant event mentions in chinese argument extraction. In: Proceedings of ACL 2013, pp. 1477–1487 (2013)
3. Narayanan, S., Harabagiu, S.: Question answering based on semantic structures. In: Proceedings of COLING 2004, pp. 693–702 (2016)
4. Parikh, A.P., Tackstrom, O., Uszkoreit, J.: A decomposable attention model for natural language inference. In: Proceedings of EMNLP 2016, pp. 2249–2255 (2016)
5. Cybulska, A., Vossen, P.: Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In: Proceedings of LREC 2014, pp. 4545–4552 (2014)
6. Bejan, C.A., Harabagiu, S.: Unsupervised event coreference resolution. Comput. Linguist. **40**(2), 311–347 (2014)
7. Ng, V., Cardie, C.: Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In: Proceedings of COLING 2002, pp. 1–7 (2002)
8. Chen, Z., Ji, H.: Graph-based event coreference resolution. In: Proceedings of TextGraphs 4, pp. 54–57 (2009)
9. Liu, Z., Araki, J., Hovy, E., Mitamura, T.: Supervised within-document event coreference using information propagation. In: LREC 2014, pp. 4539–4544 (2014)
10. Lu, J., Ng, V.: Joint learning for event coreference resolution. In: Proceedings of ACL 2017, pp. 90–101 (2017)
11. Krause, S., Xu, F., Uszkoreit, H., Weissenborn, D.: Event linking with sentential features from convolutional neural networks. In: Proceedings of CoNLL 2016, pp. 239–249 (2016)
12. Haghighi, A., Klein, D.: Simple coreference resolution with rich syntactic and semantic features. In: Proceedings of EMNLP 2009, pp. 1152–1161 (2009)

13. Vilain, M., Burger, J., Aberdeen, J., Connolly, D., Hirschman, L.: A model-theoretic coreference scoring scheme. In: Proceedings of MUC-6, pp. 45–52 (1995)
14. Mitamura, T., Liu, Z., Hovy, E.: Overview of TAC-KBP 2015 event nugget track. In: Proceedings of TAC 2015 (2015)
15. Lu, J., Ng, V.: UTD's event nugget detection and coreference system at KBP 2016. In: Proceedings of TAC 2016 (2016)