Coreference Resolution for Cybersecurity Entity: Towards Explicit, Comprehensive Cybersecurity Knowledge Graph with Low Redundancy

Zhengyu Liu[1], Haochen Su[1], Nannan Wang[1], and Cheng Huang[1,2(✉)]

[1] School of Cyber Science and Engineering, Sichuan University, Chengdu, China
opcodesec@gmail.com

[2] Anhui Province Key Laboratory of Cyberspace Security Situation Awareness and Evaluation, Hefei, China

Abstract. Cybersecurity Knowledge Graph (CKG) has become an important structure to address the current cybersecurity crises and challenges, due to its powerful ability to model, mine, and leverage massive security intelligence data. To construct a comprehensive and explicit CKG with low redundancy, coreference resolution (CR) plays a crucial role as the core step in knowledge fusion. Although the research on coreference resolution techniques in Natural Language Processing (NLP) field has made notable achievements, there is still a great gap in the cybersecurity field. Therefore, the paper first investigates the effectiveness of the existing CR models on cybersecurity corpus and presents CyberCoref, an end-to-end coreference resolution model for cybersecurity entities. We propose an entity type prediction network that not only helps to improve mention representations and provide type consistency checks, but also enables the model to distinguish the coreference among different entity types and thus run the coreference resolution more granular. To overcome the problem of implicit contextual modeling adopted by the existing CR models, we innovative propose an explicit contextual modeling method for the coreference resolution task based on semantic text matching. Finally, we improve the span representation by introducing lexical and syntactic features. The experimental results demonstrate that CyberCoref improves the F1 values on the cybersecurity corpus by 6.9% compared to existing CR models.

Keywords: Coreference resolution · Security intelligence · Semantic text matching · Entity type

1 Introduction

With the development of artificial intelligence technology, its application in the cybersecurity domain is now striding forward from perception intelligence to cognition intelligence. Sufficient and well-formed data helps to realize the

F. Li et al. (Eds.): SecureComm 2022, LNICST 462, pp. 89–108, 2023.
https://doi.org/10.1007/978-3-031-25538-0_6

"perception" stage, however the key to achieving the leap to the next stage is to refine and fuse multi-source, multi-dimensional, and heterogeneous data into knowledge, making it easier for further reasoning.

The huge amount of cybersecurity intelligence data including threat intelligence, vulnerability intelligence, and asset intelligence, provide solid data foundation for the development of intelligent security. Among them, threat intelligence portrays key information such as threat source, attack purpose, attack techniques and tactics, etc. Vulnerability intelligence includes information related to existing disclosed vulnerabilities such as impact system and software, its version, patch information, associated attack events, etc. Asset intelligence includes information related to internal assets such as accounts, servers, system software, defense mechanisms, etc. How to model, integrate, and update the security intelligence knowledge base to support further reasoning determines the effectiveness of intelligence data in actual cybersecurity battlefields and becomes the core problem that related works are trying to solve.

Knowledge Graph, as its powerful ability to correlate and fuse multi-source heterogeneous data, as well as to support precise semantic retrieval and intelligent inference analysis, has become the optimal solution for current security intelligence carriers. Existing research on Cybersecurity Knowledge Graph (CKG) construction mainly focuses on information extraction, including steps such as entity recognition [1–7], relationship extraction [8,9], and event extraction [10]. However, there is still a gap in the study of knowledge fusion, including entity disambiguation and coreference resolution steps.

Coreference resolution is the process of linking different nouns, pronouns, noun phrases, and other expressions in a text that refer to the same entity. Those various expressions of entities are defined as mentions, which increase the flow and richness of the text, but also make it more obscure to understand. It is necessary to address the reference phenomena that commonly occur in unstructured security intelligence to extract complete and valuable knowledge. As shown in Fig. 1, coreference resolution will further improve and enrich the description of cybersecurity entities at different levels and perspectives, making the extracted entities and relationships more specific, clear, and comprehensive. In addition, it links the general and vague expression of entities to those more specific, reducing the data redundancy of the CKG and thus improving its overall quality.

Although there are extensive studies on coreference resolution in the NLP field, the challenges when running coreference resolution on articles involving cybersecurity domain specific entities shouldn't be overlooked. To be more specific, by comparing the cybersecurity corpus which we constructed in this work with the general corpus dataset Ontonotes 5.0 [11], we found that: (1) cybersecurity entities are longer in length and contain more noun phrases as well as verb-object structured phrases. (2) references in cybersecurity documents have a longer distance on average. (3) cybersecurity corpus has a smaller lexicon, which results in the phenomenon of the same or looked-like spans belonging to different coreference clusters is more frequent. (4) There are more domain-specific words, abbreviations, and aliases in the cybersecurity corpus. In terms of approach, the

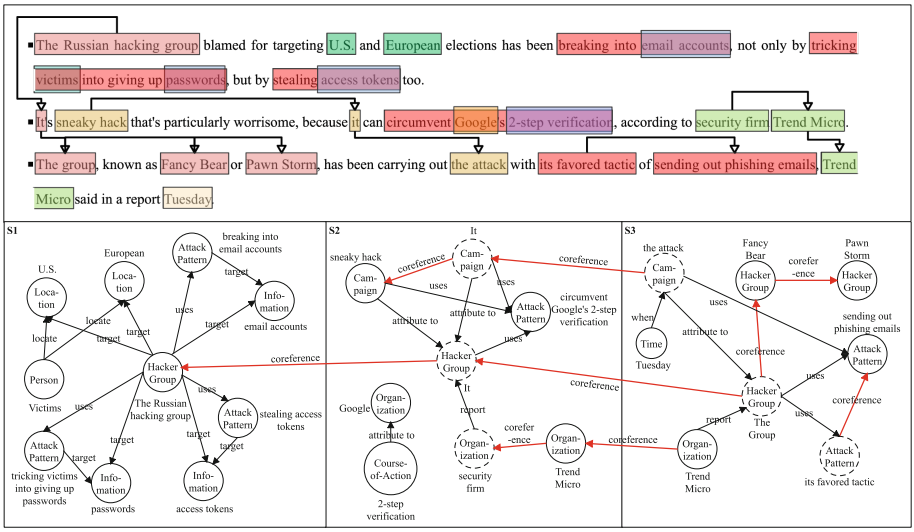

Fig. 1. Motivating example of coreference resolution on cybersecurity entities

existing state-of-the-art coreference resolution models heavily rely on BERT or its variants which are pre-trained on the large-scale general corpus and do not present expected performance when applied to domain-specific corpus [12–14]. Therefore, considering the above challenges, we need to review and evaluate the effectiveness of existing models on cybersecurity corpus and then further propose the best coreference resolution model for cybersecurity entities, targeting the characteristics of cybersecurity corpus.

Overall, our work's main contributions can be summarized as follows:

- We present CyberCoref, a document-level end-to-end coreference resolution model for cybersecurity entities, that can identify and cluster the referring cybersecurity entities within unstructured security intelligence reports in different kinds of grammatical forms including pronouns, noun phrases, verb-object structures, security domain-specific structures, and etc.
- The paper proposes a type prediction network to introduce entity-type information which enables the model to improve mention representation and provide type consistency check between mention pairs. In addition, the entity-type information enables the model to distinguish the coreference relationship of different entity types and thus perform the coreference resolution task more granular.
- To overcome the problem of implicit contextual modeling adopted by the existing coreference models, we innovative propose an explicit contextual modeling method for coreference resolution task based on semantic text matching. It uses convolutional neural networks to extract the interaction information between utterances so as to emphasize the semantic relevance of the mentions' corresponding sentences. Besides, to resolve mentions with long expressions and complex syntactic structures, CyberCoref uses an additive

attention mechanism to incorporate lexical and syntactic features for head-word finding in span representations and help the model learn more general linguistic patterns.
– To validate the effectiveness of the CyberCoref, we collected and labeled a total of 536 documents including vulnerability disclosures, APT reports, and security-related news. The proposed dataset contains 43271 cybersecurity entities, 48745 coreference links, and 6657 coreference clusters.

The rest of this paper is organized as follows: Sect. 2 presents the related work. Section 3 presents the baseline model and the details of the three key improvements proposed in this paper. Section 4 shows the dataset construction, experimental setup, the comparison results of our approach and existing coreference models on cybersecurity corpus, and the ablation study. Section 5 provides a qualitative analysis of the proposed CyberCoref to demonstrate our model's strengths and limitations. The last section concludes this paper and proposes future work.

2 Related Work

In this section, we first review the recent landmark works on neural network-based coreference resolutions in the NLP field, and then analyze the research on coreference resolution in the cybersecurity domain.

Coreference Resolution. In recent years, adopting the idea of representation learning, the neural network-based coreference resolution models have replaced the traditional machine learning models on manual feature extraction, achieving better results in datasets such as GAP [15] and OntoNotes [11] used in the Conll-2012 shared task. Wiseman et al. 2015 [16] proposed the idea of using neural networks to learn a better feature representation for mention extraction and coreference resolution on the basis of the manually extracted features. Then, to bring in coreference cluster features, Wiseman et al. 2016 [17] used recurrent neural networks to learn the global representation of entity clusters. Similarly, Clark and Manning 2016 [13] used pooling operations to generate feature representations of referring cluster pairs based on mention pair features. The great milestone work of Lee et al. in 2017 [18] completely discarded hand-extracted features and instead used word embedding models as well as deep neural networks to generate feature representations based on the idea of representation learning. The proposed mention-ranking architecture, the objective function, and the representation of mentions and mention pairs in this work were all accepted, followed by numerous subsequent works on coreference resolution [19–22]. In 2018, Lee et al. [19] accomplished two important improvements to their previous model: the introduction of higher-order inference and the coarse-to-fine pruning algorithm. The former takes the idea of the entity-level coreference resolution framework and imports global information about the coreference cluster to the mention representation. The latter improves the accuracy of candidate antecedent filtering with bearable computational complexity and memory space occupation.

As the large-scale pre-trained model BERT swept various NLP tasks as the best model in 2018, Joshi et al. 2019 [20] used BERT instead of the original word embedding and BiLSTM-based context extraction methods to generate span representation with a substantial improvement in performance on the baseline dataset. Due to the importance of span representation in the coreference resolution task, Joshi [21] released SpanBERT in the same year which is more suitable for span boundary sensitive tasks, and achieves better results compared to the original BERT model. In addition, the corefBERT from Ye et al. 2020 [22], which uses coreference resolution as the self-training task of the BERT model, also has excellent performance.

The word-level coreference resolution model was proposed by Kirstain et al. in 2021 [23]. Dobrovolskii [24] inherited the idea from Kirstain et al. which is to accomplish the task from the word level rather than the span level, achieving similar results to the span-based coreference resolution model on the baseline datasets. The word-level model has the advantages of less search space in the coreference resolution step and avoidance of incorrect pruning in the mention detection step. However, using word embeddings directly instead of span representations will lead to missing certain information, especially when dealing with long and complex mentions. To evaluate the effectiveness of the word-level model in the cybersecurity domain, we compared their model in Sect. 4.

Coreference Resolution in the Field of Security. Although topics such as information extraction and knowledge graph construction [1–10, 25, 26] have been widely studied within the cybersecurity domain in recent years, research on coreference resolution in the cybersecurity domain is still relatively scarce. Hu et al. [27] modeled the coreference resolution task jointly with the relation extraction, treating the coreference relation as one of the inter-entity relation types. Zhang et al. [28] first extracts mentions from a given document by using a sequence labeling neural network, and then applies a random forest algorithm with custom rules to complete the resolution of extracted mentions. However, it should be noted that their works consider the coreference resolution as a cascading task of entity recognition that may suffer from error propagation. And also, their approaches regarding mention detection as a sequence labeling task are not able to cope with the nested entities natively, which would result in low recall rates.

3 Methodology

In this work, we used the model proposed in Joshi et al. 2019 [20] as our baseline model, which has achieved outstanding results on the OntoNotes 5.0 general corpus. The model adopts the higher-order inference as well as the coarse-to-fine pruning algorithm proposed by Lee et al. [18] and continues the classic task modeling, learning objectives, score architecture, and span representation proposed by Lee et al. 2017 [13]. Subsection 3.1 will focus on an overview of our baseline model.

As discussed in Sect. 1, running coreference references on the cybersecurity corpus is going to face more challenges than the general corpus. So, in order to better adapt to the characteristics of cybersecurity entities, we make the following improvements (Fig. 2) to the baseline model. First, to avoid the over-reliance on the similarity of words, lexical and syntactic features are introduced in the span representation to help the model learn more general language rules and improve its generalization ability, as detailed in Sect. 3.2. Secondly, inspired by the fact that human beings would significantly narrow down the search space by sifting out the mentions in the irrelevant sentences when finding the most appropriate antecedent, this work proposes an explicit contextual modeling network based on semantic text matching, as detailed in Subsect. 3.3. Finally, considering that different entity types do not share the same degree of reference matching, this work proposes an entity-type prediction network to keep the model aware of entity types in both mention detection and coreference resolution steps, as detailed in Subsect. 3.4.

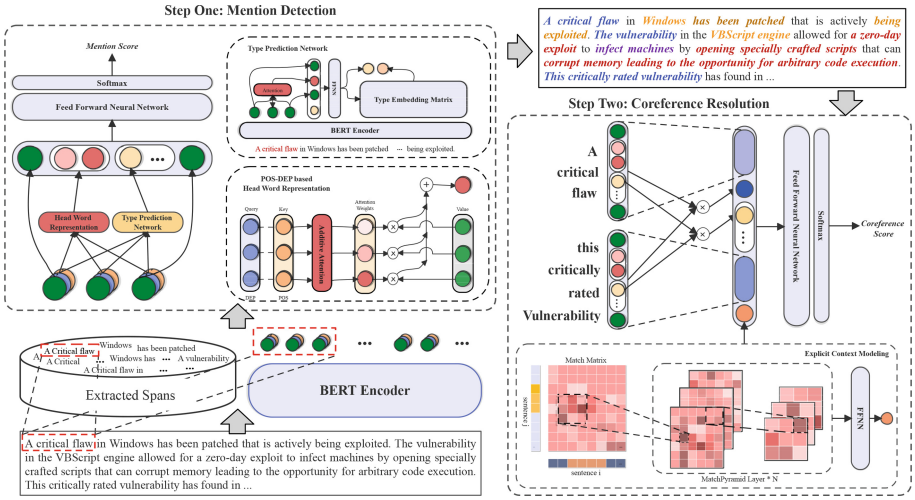

Fig. 2. Overview of CyberCoref architecture.

3.1 Baseline

The modeling for the coreference resolution task is shown below. Given a document D containing T words, it corresponds to $N = T(T+1)/2$ spans. For all spans i $(1 \leq i \leq N)$, each of them could find an antecedent $y_i \in Y(i) = \{\epsilon, 1, 2, \ldots, i-1\}$. If the span i corresponds to a dummy antecedent ϵ, it means that the span is not a mention, or the mention does not have a corresponding antecedent.

After encoding text segments by the BERT model, we can get the embedding of each token in the span i. Based on that, we can get the representation of the

span i, i.e. g_i, including embeddings of its start and end positions, respectively, headword representation generated by an attention mechanism, and the span width feature. Through the mention score function $s_m(\cdot)$, we will get the score used to determine whether the span is a mention or not.

$$s_m(i) = FFNN_m(g_i) \tag{1}$$

For the spans i and j as the extracted mention and its candidate antecedent, respectively, we follow the coarse-to-fine pruning algorithm [18], using a simplified version of the coreference scoring function $s_c(\cdot)$ and a more precise coreference scoring function $s_a(\cdot)$ to determine whether there is a coreference relationship between them. For coreference scoring, the representation of the mention pairs contains g_i, g_j, $g_i \circ g_j$ and other features including representation of the distance between them.

$$s_c(i,j) = g_i^\top W_c g_j \tag{2}$$

$$s_a(i,j) = FFNN_a([g_i, g_j, g_i \circ g_j, \phi_a]) \tag{3}$$

where W_c is a learned weight matrix, $\cdot$ denotes the dot product, $\circ$ denotes element-wise multiplication, and $FFNN$ denotes a feed-forward neural network that computes a nonlinear mapping from input to output vectors.

When it comes to specific inference, a three-step pruning will be performed to ensure computational efficiency. To start with, mentions will be filtered from all spans by using the unary function $s_m(\cdot)$. Then, for each extracted mention, we select top K mentions as its candidate antecedents based on the score $s_m(i)+s_m(j)+s_c(i,j)$, and finally use the function $s_a(i,j)$ to achieve refined coreference scoring.

3.2 Combining Lexical and Syntactic Features

Suggested by the work [18] that generating the headword representation from word embedding vectors would still be error prone. Therefore, to more correctly represent those headwords in span representation, we use an additive attention mechanism based on lexical and syntactic dependency features instead of the original attention mechanism which only based on words themselves.

Given a span i with a length l, the corresponding word embedding, part-of-speech embedding and syntactic dependencies embedding are $S_i \in \mathbb{R}^{l \times d_{word}}$, $P_i \in \mathbb{R}^{l \times d_{pos}}$ and $R_i \in \mathbb{R}^{l \times d_{deprel}}$, respectively. The additive attention scoring function a and attention weights α based on lexical and syntactic dependency features are shown below. Where the lexical embedding matrix and word embedding matrix will be separately used as the key K and the value V of attention, and the syntactic role embedding matrix will be used as the query Q.

$$a_t = W_v^\top tanh(w_p p_t + w_r r_t) \tag{4}$$

$$\alpha_t^i = \frac{exp(a_t)}{\sum_{k=START(i)}^{END(i)} exp(a_k)} \tag{5}$$

where $w_p \in \mathbb{R}^{h \times d_{pos}}$ and $w_r \in \mathbb{R}^{h \times d_{deprel}}$ are the learnable weight matrices and $\alpha_{i,t}$ is the attention weight of the token corresponding to the position t in span i. Therefore, obtained by the attention mechanism, the final headword representation of span i, i.e. $\widehat{s_i}$, which incorporates part-of-speech information and syntactic dependencies, is shown below.

$$\widehat{s_i} = \sum_{t=START(i)}^{END(i)} \alpha_t^i \cdot s_t^i \tag{6}$$

In addition, mean pooling of embedding of part-of-speech and syntactic features of the span are taken separately, and we concatenate them with the embedding of type information (detailed in Subsect. 3.4) and length of span to get the feature vector ϕ_m. In summary, the representation of the span i is shown below.

$$\phi_m(i) = [F_{type(i)}, F_{width(i)}, F_{pos(i)}, F_{deprel(i)}] \tag{7}$$

$$g_i = [s^i_{START(i)}, s^i_{END(i)}, \widehat{s_i}, \phi_{m(i)}] \tag{8}$$

3.3 Explicit Contextual Modeling

The baseline model uses only implicit contextual modeling, i.e., it relies on the contextual semantic word embeddings generated by the pre-trained model to reflect the relevance of encoded segments. However, we note that humans will first screen out candidate antecedents from completely irrelevant or conflicting statements and keep a small set of mentions that appear in closely related sentences based on context relevance. This intuitive idea is consistent with the fact that cybersecurity documents often repeatedly present descriptions of events, exploits, vulnerabilities, and attackers from different perspectives and degrees. Therefore, the relevance of context can become a very important factor in the task of coreference resolution. We aim to learn from semantic text matching models to explicitly model a closer context of mentions and their candidate antecedents, to determine whether their contexts have the same discussion objects or convey similar meanings.

Due to the powerful semantic modeling capability of the BERT model, the word embedding vector contains sufficient information for determining the semantic similarity of sentence pairs. Therefore, a simple and effective way is used in this paper to extract the local similarity features from the token-level matching matrix which can reflect the correlation between utterances. The main network structure refers to the MatchPyramid proposed by the work of Pang et al. [29], which uses a hierarchical CNN network to extract local information at a different level and introduces a dynamic pooling mechanism to handle pairs of sentences with different lengths.

For the extracted mention i and the candidate antecedent j, we can get the corresponding sentence S_i and S_j with the length n and m, respectively. The sentence representation $S_i = \{s_1^i, s_2^i, \ldots, s_n^i\}$ and $S_j = \{s_1^j, s_2^j, \ldots, s_m^j\}$, as well

as the initial matching matrix M, can be obtained based on the embedding of each token. Calculation of each position of the initial matching matrix is shown as follows, where $Sim(\cdot)$ represents the word similarity score function, which can be dot product or cosine similarity. Finally, the MatchPyramid will be applied to extract features from the matching matrix and is followed by a feed forward network to get the fixed length $F_{sent-pair}$ feature which represents the contextual relevance.

$$M_{tk} = Sim(s_t^i, s_k^j) \tag{9}$$

$$F_{sent-pair} = FFNN_{context}(MatchPyramid(M)) \tag{10}$$

3.4 Entity Type Information

In the process of mention detection, for the span i and the embedding X_i of the sentence where i is located encoded by the BERT model, we can get its corresponding embedding vector $\{x^i_{START(i)}, ..., x^i_{END(i)}\}$. Similarly, we use the idea of mention representation to predict the corresponding type of each span. The span representation h_i used for type prediction consists of the embedding of the start and end token, the headword representation fused by the attention mechanism, and the feature to encode the span width. We use the embedding X_i that represents a closer context (i.e., the sentence embedding) instead of the segment embedding S_i to make the network more convenient for the pre-training on the type prediction subtask, which helps the network learn to classify the types of span.

$$a_t = FFNN_\alpha(X_i) \tag{11}$$

$$\alpha_t^i = \frac{exp(a_t)}{\sum_{k=START(i)}^{END(i)} exp(a_k)} \tag{12}$$

$$\widehat{x_i} = \sum_{t=START(i)}^{END(i)} \alpha_t^i \cdot x_t^i \tag{13}$$

$$h_i = [x^i_{START(i)}, x^i_{END(i)}, \widehat{x_i}, F_{width(i)}] \tag{14}$$

Next, we use a feed forward neural network to score the likelihoods of different types. For the type given the highest possibility, we also get its embedding $e_{type(i)} \in \mathbb{R}^{d_{type}}$.

$$t_i = FFNN_{type}(h_i) \tag{15}$$

$$e_{type(i)} = Embedding_{type}(argmax(t_i)) \tag{16}$$

$$F_{type} = [t_i, e_{type(i)}] \tag{17}$$

During the coreference resolution, to provide an additional check of mention-pair type consistency, we add their type embeddings as well as the cosine similarity of their type likelihood scores to the mention-pair representation. Finally, ϕ_a is calculated as shown below:

$$\phi_a(i,j) = [F_{distance}, F_{sent-pair}, e_{type(i)}, e_{type(j)}, cos(t_i, t_j)] \tag{18}$$

To help the network generate the correct entity types based on the span and the given context, we pre-trained it with the entity type prediction subtask. The training samples are obtained from the mention detection steps of the baseline model, including the mentions and their corresponding sentences. The correct mentions are labeled with their corresponding entity types, while the wrongly extracted mentions are labeled as None. After sufficient training on this multi-classification subtask, the entity type prediction network can better distinguish between true and false mentions with similar boundaries and assign the correct entity type.

4 Evaluation

In this section, we will design a series of experiments to demonstrate the effectiveness and superiority of our proposed model CyberCoref especially towards the cybersecurity entity coreference resolution. Firstly, we will provide a comprehensive introduction to our dataset. Then, we clarify the experiment setup and show the results of CyberCoref and other representative coreference models in the NLP field on our dataset. Finally, we present a thorough ablation study on the proposed networks of CyberCoref.

4.1 Dataset

The experimental dataset is derived from publicly available security intelligence, including vulnerability disclosures[1], APT reports[2], and security-related news [10]. Our work refers to the ontology construction of UCO [30] and the threat intelligence sharing framework STIX2.1 [3] and redefines 29 cybersecurity entity types. And then we manually annotated 43,271 cybersecurity entities and 48,745 intra-document coreference links, which construct 6,657 coreference clusters, in 536 security-related articles. The specific entity definitions and annotation guidelines are detailed in our open source repository[4].

Regarding the data processing, we first extract plain text from rich text documents (e.g. pdf or html), and remove all embedded images, tables, and inserted code segments. Then we perform a simple data cleaning by replacing all non-ascii encoded characters with spaces and rewrite the protected forms from clicks by mistake of IP addresses, email addresses, and web addresses. Finally, we segment the long articles to ensure the best performance of the model.

The annotation of the dataset is conducted on the Brat platform [31]. The dataset annotation processes are as follows: firstly, we annotated the entities according to the defined entity types. Then, we annotate the coreference relations between the annotated entities, at the same time, proofread our previous annotation. The whole annotation work was done by two graduate students and one senior undergraduate student, which all major in cybersecurity.

[1] https://github.com/pburkart/Vulnerability-Research-Blogs.
[2] https://github.com/CyberMonitor/APT_CyberCriminal_Campaign_Collections.
[3] https://oasis-open.github.io/cti-documentation/stix/intro.
[4] https://github.com/jackfromeast/CyberCoref.

4.2 Evaluation Setup

The CyberCoref model is implemented with reference to the Bert-based c2f-coref model proposed by Joshi et al. 2019 [20], which is now open-sourced to the Github repository.

Model Architecture. We take the SpanBERT for word embedding, which is a pre-trained model from Hugging Face[5] at the based size with an embedding dimension of 768. We use the CoreNLP[6] for both part-of-speech and syntactic features extraction, and they all have an embedding dimension of 64. The embedding dimension of both entity type and context relevance features is 64 as well, and the distance and width features are binned into the following buckets [1, 2, 3, 4, 5–7, 8–15, 16–31, 32–63, 64–127, 128–255, 256–511, 512–1023, 1024+] and then embedded as 16-dimensional size vectors. The number of MatchPyramid layers is 2, the convolution kernel sizes are 5×5 and 3×3, the pooling layer sizes are 10×10 and 5×5, and the numbers of feature maps are 8 and 16, respectively. The hidden layer sizes of the feedforward neural networks $FFNN_m$ and $FFNN_a$ for mention and coreference scoring are both 1024, the hidden layer sizes of $FFNN_\alpha$ and $FFNN_{type}$ in the entity type prediction network are 512 and 1024, respectively, and in the context relevance representation network $FFNN_{context}$, the hidden layer size is 128. We adopt LeakyReLU [32]as the activation function, and the dropout rate is set to 0.3.

Inference. The pruning threshold λ is set to 0.3, the maximum span length is 20. For each extracted mention or word, the number $K = 50$ of candidate antecedents are going to be selected. Referring to the experimental results in the work of Joshi et al. [20], the maximum segment length is set to 384 for word embedding. We use the higher-order inference algorithm based on antecedent distribution proposed by Lee et al. 2018 [19], and the number of iteration rounds is set to 2.

Learning. The optimizer for the models is AdamW, with a learning rate of 1e-5 for the pre-training weights of the BERT model and 3e-4 for the rest of the network structures. Such a learning rate setting allows the optimization of the models as a whole to be adjusted to the optimal position simultaneously. The training batch size for all models is 1, i.e., one article at a time. The models are all trained for up to 60 epochs. The training and validation sets are randomly divided in a 4:1 ratio. In addition, to avoid any degree of data leakage, training samples for the entity type prediction subtask are all from the training set.

[5] https://huggingface.co/.
[6] https://stanfordnlp.github.io/CoreNLP/.

Compared Models. For the replicated models used for comparison, the model architecture hyperparameters are set with reference to their original papers, and the inference parameters and training parameters are the same as above, except for the following notes. For the e2e-coref model proposed by Lee et al. 2017 [18], the optimizer is Adam with a learning rate of 3e-4. The word-level coreference resolution model wl-coref [24] also uses based size SpanBERT to complete word embedding, and the maximum segment length is set to 512. The remaining BERT-based span-level models [20–22] are all set to a maximum segment length of 384, and all use based size pre-trained models.

4.3 Coreference Results

Table 1 and Table 2 demonstrate the performance of CyberCoref and compared coreference resolution models on our cybersecurity corpus. The evaluation metrics are the MUC, B-Cubed, and $CEAF_{\phi_3}$ which were used in the conll-2012 coreference resolution shared task [11], and the LEA [33] evaluation metrics proposed by Moosavi et al. in 2016. Since these models heavily rely on span representation, they have high requirements for word embeddings. Models [20–24] using large-scale pre-trained dynamic word embeddings significantly perform better than using the embedding method combined GloVe, Turian, and Char-CNN [18]. The word-level model shows limited ability to distinguish long and complex spans, displaying lower scores compared to span-based models. In general, due to the vast differences between datasets, models that perform well on general corpora, like OntoNotes 5.0 and GAP, do not achieve the expected results on our dataset which is far more challenging. In contrast, the proposed CyberCoref achieves better results in the following four evaluation metrics and becomes the best coreference resolution model for cybersecurity entities.

Table 1. Results on the validation set of our cybersecurity corpus. The final column (Avg. F1) is the main evaluation metric, computed by averaging the F1 of MUC, B-cubed, and $CEAF_{\phi_3}$

	MUC			*B − cubed*			$CEAF_{\phi_3}$			
	Prec.	Rec.	**F1**	Prec.	Rec.	F1	Prec.	Rec.	F1	Avg. F1
Lee et al. 2017 [18]	9.8	**58.8**	16.7	2.0	**81.6**	4.9	4.0	18.2	6.7	9.4
Joshi et al.2019a [20]	62.2	27.6	36.2	**80.7**	63.7	69.4	65.5	29.2	38.6	48.1
Joshi et al.2019b [21]	56.4	38.9	43.6	74.2	72.0	71.7	59.8	39.0	45.0	53.4
Ye et al. 2020 [22]	59.2	35.9	42.4	80.3	68.1	72.3	61.5	37.7	44.7	53.1
Kirstain et al. 2021 [24]	24.5	23.3	22.8	61.3	62.4	61.0	31.2	28.6	28.8	37.5
Proposed Model	**63.8**	47.9	52.6	79.0	76.5	**76.6**	**66.5**	**47.1**	**53.3**	**60.9**

In terms of mention detection, the span-based model pursues a high recall rate so that as many mentions as possible are selected in a fixed number of extracted spans. While the selected non-mention spans and singletons will be screened in

the following coreference resolution step later. As shown in Table 3, our proposed model achieves the best result for the mention detection by increasing in both recall and precision rates.

Table 2. Results on the validation set of our cybersecurity corpus with the LEA metric.

	LEA		
	Prec.	Rec.	F1
Lee et al. 2017 [18]	0.3	**49.3**	0.6
Joshi et al. 2019a [20]	**56.0**	19.6	26.2
Joshi et al. 2019b [21]	47.1	30.6	33.4
Ye et al. 2020 [22]	52.6	26.7	32.2
Kirstain et al. 2021 [24]	17.7	17.4	16.1
Proposed model	54.7	39.0	**42.0**

Table 3. Results of the mention detection step.

	Prec.	Rec.	F1
Lee et al. 2017 [18]	15.6	65.9	25.2
Joshi et al. 2019a [20]	27.1	87.9	41.4
Joshi et al. 2019b [21]	26.8	87.8	41.1
Ye et al. 2020 [22]	27.1	88	41.4
Kirstain et al. 2021 [24]	–	–	–
Proposed model	**28.3**	**92.1**	**43.3**

4.4 Ablations

Lexical and Syntactic Features. The span representation plays a crucial role as the basis for the solution of both the mention detection and coreference resolution task. Four different ways of encoding part-of-speech and syntactic features are used as comparisons in this experiment. Due to the different token lengths of spans and each token corresponds to a separate part-of-speech and syntactic dependency label, this problem can be viewed as a variable-length category feature sequence encoding problem. The first and most straightforward idea is to embed the two features separately using the EmbeddingBags, and then average or sum the corresponding lexical and syntactic feature embeddings before concatenating them together. In the third approach, a Long short-term memory (LSTM) network is used for feature extraction of variable-length feature sequences, where the hidden state of the last non-padded time step is taken as the feature embedding. The last approach is our proposed additive attention mechanism, which is introduced in Sect. 3.

Table 4. Ablation study of the proposed architectures in CyberCore, where MD and CR stand for the mention detection and coreference resolution, respectively.

		MD		CR	
		Rec.	Prec.	Conll-2012 Avg. F1	LEA F1
	Baseline	87.8	26.8	53.4	33.4
Lexical & syntactic features	+EmbeddingBags-mean	86.5	26.4	45.6	29.4
	+EmbeddingBags-sum	76.5	23.3	42.1	21.1
	+LSTM	85.6	26.5	41.6	29.8
	+AttitiveAttention	**88.3**	**27.2**	**56.0**	**36.0**
Context modeling	+BiGRU	87.4	27.0	53.1	33.4
	+BiLSTM	87.3	26.9	46.0	19.8
	+Cos-MaxPooling	**87.8**	26.4	46.3	20.3
	+Cos-MatchPyramid [29]	87.7	26.5	53.6	34.0
	+Cos-Dot-MatchPyramid [29]	87.7	**26.8**	**55.6**	**36.8**
Entity type prediction	+Golden types	95.5	29.4	70.9	59.2
	+standalone-TPM	90.5	27.4	56.2	36.2
	+E2E-TPM with pre-trained weights	**92.0**	**28.4**	**59.9**	**40.2**

The experimental result shows that the straightforward introduction of part-of-speech and syntactic features does not help the selection of mentions, while using the additive attention mechanism to introduce lexical and syntactic features for headword finding can better identify the most appropriate headword within a span and help its representation, thus improving the effectiveness of mention extraction and coreference resolution tasks in terms of precision rate.

Context Relevance Modeling. We come up with five explicit context modeling approaches to compare with the baseline model of Joshi et al. 2019b [21]. The first two approaches are based on recurrent neural networks, which encode the concatenated sentences where the mention and its candidate mentions are located in and take the hidden state at the last time step as the contextual feature of the two sentences. The last three approaches use matching matrices generated by cosine similarity or dot product function, which can better demonstrate the correlation between tokens than recurrent neural networks. The experimental results show that using MatchPyramid to extract features from the Cos and Dot matching matrices can demonstrate the relevance of sentences more precisely and provides valuable information for the coreference resolution of selected mentions.

Entity Type Prediction. The performance of different networks on the entity type prediction subtask is shown in Table 5. The previous works [34,35] demonstrated the noteworthy improvement of adding special tags (e.g., ⟨tag⟩ and ⟨/tag⟩) before and after the target span on span-related tasks such as entity typing and relationship extraction. Therefore, we investigate the effect of adding the boundary tags and compare the two proposed ways of span representation, i.e. concatenating the corresponding embedding of pre-and-post tags or performing average pooling for spans. The experimental result shows that tags can help

the model better perceive the boundaries of span and grasp the semantics of content within the span tags, thus the concatenation of simple tag embedding can also lead to a good enough representation of spans. However, since the embedding of tags has not been pre-trained by BERT, its upper limit is inferior to the span representation proposed in this work, which consider the span boundary and overall content.

We then compare two ways of incorporating pre-trained entity type prediction network to CyberCoref: as a stand-alone model without participating in the parameter update of the overall model training to ensure the accuracy on the type prediction subtask; as part of the end-to-end model participating in the parameter update, sharing the weights of the BERT model which are pre-trained on the subtask of entity type prediction with the overall model. In addition, the performance of using exactly the right entity types (golden types) is also shown in Table 4. The result shows that the introduction of the entity type prediction network with pre-trained weights to initialize the end-to-end model works best.

Table 5. Performance of different networks on the entity type prediction subtask.

		Type Prediction			
		Prec.	Rec.	Micro-F1	Weighted-F1
Tagged [34]	+tag [34]	81.9	76.8	76.8	77.9
	+mean [35]	80.4	76.7	76.7	77.1
	+proposed network	81.9	76.8	76.8	77.9
Original	+mean	79.8	75.8	75.9	76.3
	+proposed network	83.4	78.3	78.3	79.4

5 Analysis

Strengths. After manually analyzing the errors on a set of validation samples, we found that for the more common and less error-prone coreferences, such as coreferences of pronouns or noun phrases in the same sentence or adjacent sentences, our model can determine them accurately based on the training with a large number of similar patterns and the guidance of entity types. The introduction of part-of-speech and syntactic features is helpful for the representation of longer mentions in the form of verb-object or other more complex structures. For example, in *(Subject)-Verb-Object sentence pattern* shown in Table 6, for Attack-Pattern, which are common in cybersecurity corpora and are usually used to describe the attack process or represent the attack features, CyberCoref can identify these mentions and complete the coreference resolution among them correctly. Furthermore, in APT or other cyber attack reports, we find that different parts of the article have obvious distinctions in discussion objects, as *Multi-discussion Objects in the same passage* shown in Table 6. The explicit sentence-based contextual modeling proposed in this work better reflects the relevance

of sentences than the original segment-based implicit contextual semantic modeling. In this case, although the mention "new ransomware" appears twice, the key signals in the sentence such as "In other ransomware news" and the words interaction between their contexts help the model cluster them to different coreference groups rather than the same one.

Weaknesses. However, the coreference in the cybersecurity corpus is more complex than expected, and there are many challenging error-prone scenarios that make our model CyberCoref not always reliable. Many same or similar words that are commonly used in cybersecurity topics may refer to different entities, such as "vulnerability", "issue", "company", "attack", etc. This is similar to that of pronouns but the former is more difficult to handle, as these across coreference clusters high-frequency words are usually not constrained by distance but depend only on semantic expression. For example, as in the cases in *Same or Look-alike Strings*, CyberCoref is highly susceptible to the similarity of the words themselves, resulting in false positive resolutions, thus causing greater degradation in the evaluation metrics. In addition, the presence of many comparisons and citation descriptions in the cybersecurity corpus, some paragraphs will have the phenomenon of discussing multiple entities of the same type, making coreference resolution further difficult.

6 Conclusion and Future Work

In conclusion, we explore the effectiveness of existing coreference resolution models on cybersecurity corpus. To address the limitations of their performance, we propose CyberCoref, a document-level end-to-end coreference resolution model for cybersecurity entities. Based on the three improvements proposed in this work, including entity type prediction networks, explicit contextual modeling, and the introduction of lexical and syntactic features, CyberCoref improves the average F1 value of the four evaluation metrics by 6.9% on the dataset constructed in this work. However, when it comes to more complex coreference expressions, CyberCoref still has much room for improvement. In our future work, we will focus on solving the challenging coreference cases mentioned in Sect. 5.

Acknowledgment. This research is funded by the National Natural Science Foundation of China (No.61902265), National Key Research and Development Program of China (No.2021YFB3100500), Open Fund of Anhui Province Key Laboratory of Cyberspace Security Situation Awareness and Evaluation (No. CSSAE-2021-001).

A Challenging Coreference Cases

Table 6. Challenging coreference cases in our cybersecurity corpus. For better illustration, we only mark up the typical coreferences that reflects the displayed coreference types shown in the left side. Manual labels and results given by CyberCoref are shown in the right side.

Challenging types	Examples	Results
Same or Look-alike Strings	This is only the latest exploit to hit Adobe Flash - earlier in June, **a zero-day Flash vulnerability(1)** was is being exploited in the wild in targeted attacks against Windows users in the Middle East, according to researchers. Adobe dealt with **another zero-day Flash vulnerability(2)** back in February, which was exploited by North Korean hackers	Golden: [(1)], [(2)] CyberCoref: [(1), (2)]
	Impacted is Adobe Flash Player Desktop Runtime, **Adobe Flash Player(1)** for Google Chrome; **Adobe Flash Player(2)** for Microsoft Edge and Internet Explorer 11; all for versions 31.0.0.153 and earlier	Golden:[(1)], [(2)] CyberCoref:[(1), (2)]
(Subject)-Verb-Object pattern	The only problem is that detecting either the hacked bank or the hacked ATM is almost impossible as most of the malicious behavior takes place via self-deleting malware and malicious PowerShell scripts executing in memory, **without leaving any artifacts on disk(1)**. Once the bank server/computer or the AMT is rebooted, most of **the clues are wiped from memory(2)**	Golden: [(1), (2)] CyberCoref: [(1), (2)]
	Microsoft Windows users beware of an unpatched memory corruption bug which could **be exploited to cause denial of service (DoS) attacks(1)** as well as other exploits If a user connects to a malicious SMB server, a vulnerable Windows client system may **crash and display a blue screen of death (BSOD) in mrxsmb20.sys(2)**, the advisory said	Golden: [(1), (2)] CyberCoref: [(1), (2)]
Multi-discussion Objects in the same paragraph	**Israeli mobile forensics firm(1) Cellebrite(2)** has announced that **it(3)** has suffered a data breach following an unauthorized access to an external web server. The confirmation comes a few hours after **Motherboard(4)** released general information about 900 GB of data that **they(5)** obtained and has supposedly been stolen from **the firm(6)**. The cache includes alleged usernames and passwords for logging into Cellebrite databases connected to **the company(7)**'s my.cellebrite domain, the publication noted	Golden: [(1), (2), (3), (6), (7)], [(4), (5)] CyberCoref: [(1), (2), (3), (7)], [(4), (5), (6)]
	This vulnerability(1) has been assigned **the CVE-2018-17456 ID(2)** and is similar to a previous **CVE-2017-1000117(3) option injection vulnerability(4)**. Like **the previous vulnerability(5)**, a malicious repository can create a .gitmodules file that contains an URL that starts with a dash	Golden: [(1), (2)], [(3), (4), (5)] CyberCoref: [(1), (2)], [(3), (4), (5)]
Multi-discussion Objects in the same passage	One tried-and-true technique continues to be hiding malware inside fake versions of popular files, then distributing those fake versions via app stores. To wit, last week researchers at the security firm ESET spotted **new ransomware(1) - Filecoder.E(2)** - circulating via BitTorrent, disguised as a"patcher" that purports to allow Mac users to crack such applications as Adobe Premiere Pro CC and Microsoft Office 2016 ... In other ransomware news, **new ransomware(3)** known as **Trump Locker(4)** - not to be confused with Trumpcryption - turns out to be a lightly repackaged version of VenusLocker ransomware, according to Lawrence Abrams of the security analysis site Bleeping Computer, as well as the researchers known as MalwareHunter Team	Golden: [(1), (2)], [(3), (4)] CyberCoref: [(1), (2)], [(3), (4)]

References

1. Jones, C.L., Bridges, R.A., Huffer, K.M., Goodall, J.R.: Towards a relation extraction framework for cyber-security concepts. In: Proceedings of the 10th Annual Cyber and Information Security Research Conference, pp. 1–4 (2015)
2. Mittal, S., Das, P.K., Mulwad, V., Joshi, A., Finin, T.: Cybertwitter: using twitter to generate alerts for cybersecurity threats and vulnerabilities. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 860–867. IEEE (2016)
3. Liao, X., Yuan, K., Wang, X., Li, Z., Xing, L., Beyah, R.: Acing the ioc game: toward automatic discovery and analysis of open-source cyber threat intelligence. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 755–766 (2016)
4. Zhu, Z., Dumitras, T.: Chainsmith: automatically learning the semantics of malicious campaigns by mining threat intelligence reports. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 458–472. IEEE (2018)
5. Ghazi, Y., Anwar, Z., Mumtaz, R., Saleem, S., Tahir, A.: A supervised machine learning based approach for automatically extracting high-level threat intelligence from unstructured sources. In: 2018 International Conference on Frontiers of Information Technology (FIT), pp. 129–134. IEEE (2018)
6. Zhao, J., Yan, Q., Li, J., Shao, M., He, Z., Li, B.: Timiner: automatically extracting and analyzing categorized cyber threat intelligence from social data. Comput. Secur. **95**, 101867 (2020)
7. Husari, G., Niu, X., Chu, B., Al-Shaer, E.: Using entropy and mutual information to extract threat actions from cyber threat intelligence. In: 2018 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 1–6. IEEE (2018)
8. Guo, Y., et al.: CyberRel: joint entity and relation extraction for cybersecurity concepts. In: Gao, D., Li, Q., Guan, X., Liao, X. (eds.) ICICS 2021. LNCS, vol. 12918, pp. 447–463. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86890-1_25
9. Pingle, A., Piplai, A., Mittal, S., Joshi, A., Holt, J., Zak, R.: Relext: relation extraction using deep learning approaches for cybersecurity knowledge graph improvement. In: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 879–886 (2019)
10. Satyapanich, T., Ferraro, F., Finin, T.: Casie: extracting cybersecurity event information from text. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 8749–8757 (2020)
11. Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., Zhang, Y.: Conll-2012 shared task: modeling multilingual unrestricted coreference in ontonotes. In: Joint Conference on EMNLP and CoNLL-Shared Task, pp. 1–40 (2012)
12. Brack, A., Müller, D.U., Hoppe, A., Ewerth, R.: Coreference resolution in research papers from multiple domains. In: Hiemstra, D., Moens, M.-F., Mothe, J., Perego, R., Potthast, M., Sebastiani, F. (eds.) ECIR 2021. LNCS, vol. 12656, pp. 79–97. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72113-8_6
13. Clark, K., Manning, C.D.: Improving coreference resolution by learning entity-level distributed representations. arXiv preprint arXiv:1606.01323 (2016)
14. Timmapathini, H., et al.: Probing the spanbert architecture to interpret scientific domain adaptation challenges for coreference resolution. In: SDU@ AAAI (2021)
15. Webster, K., Recasens, M., Axelrod, V., Baldridge, J.: Mind the gap: a balanced corpus of gendered ambiguous pronouns. Trans. Assoc. Comput. Linguist. **6**, 605–617 (2018)

16. Wiseman, S.J., Rush, A.M., Shieber, S.M., Weston, J.: Learning anaphoricity and antecedent ranking features for coreference resolution. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 1: Long Papers. Association for Computational Linguistics (2015)
17. Wiseman, S., Rush, A.M., Shieber, S.M.: Learning global features for coreference resolution. arXiv preprint arXiv:1604.03035 (2016)
18. Lee, K., He, L., Lewis, M., Zettlemoyer, L.: End-to-end neural coreference resolution. arXiv preprint arXiv:1707.07045 (2017)
19. Lee, K., He, L., Zettlemoyer, L.: Higher-order coreference resolution with coarse-to-fine inference. arXiv preprint arXiv:1804.05392 (2018)
20. Joshi, M., Levy, O., Weld, D.S., Zettlemoyer, L.: Bert for coreference resolution: Baselines and analysis. arXiv preprint arXiv:1908.09091 (2019)
21. Joshi, M., Chen, D., Liu, Y., Weld, D.S., Zettlemoyer, L., Levy, O.: Spanbert: improving pre-training by representing and predicting spans. Trans. Assoc. Comput. Linguist. **8**, 64–77 (2020)
22. Ye, D., et al.: Coreferential reasoning learning for language representation. arXiv preprint arXiv:2004.06870 (2020)
23. Kirstain, Y., Ram, O., Levy, O.: Coreference resolution without span representations. arXiv preprint arXiv:2101.00434 (2021)
24. Dobrovolskii, V.: Word-level coreference resolution. arXiv preprint arXiv:2109.04127 (2021)
25. Liu, K., Wang, F., Ding, Z., Liang, S., Yu, Z., Zhou, Y.: A review of knowledge graph application scenarios in cyber security. arXiv preprint arXiv:2204.04769 (2022)
26. Fang, Y., Zhang, Y., Huang, C.: Cybereyes: cybersecurity entity recognition model based on graph convolutional network. Comput. J. **64**(8), 1215–1225 (2021)
27. Hu, Y., Guo, Y., Liu, J., Zhang, H.: A hybrid method of coreference resolution in information security. Comput. Mater. Continua **64**(2), 1297–1315 (2020)
28. Wang, X., Xiong, M., Luo, Y., Li, N., Jiang, Z., Xiong, Z.: Joint learning for document-level threat intelligence relation extraction and coreference resolution based on gcn. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 584–591. IEEE (2020)
29. Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., Cheng, X.: Text matching as image recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30 (2016)
30. Syed, Z., Padia, A., Finin, T., Mathews, L., Joshi, A.: UCO: a unified cybersecurity ontology. In: Workshops at the Thirtieth AAAI Conference on Artificial Intelligence (2016)
31. Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: Brat: a web-based tool for nlp-assisted text annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 102–107 (2012)
32. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853 (2015)

33. Moosavi, N.S., Strube, M.: Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers), pp. 632–642 (2016)
34. Khosla, S., Rose, C.: Using type information to improve entity coreference resolution. arXiv preprint arXiv:2010.05738 (2020)
35. Soares, L.B., FitzGerald, N., Ling, J., Kwiatkowski, T.: Matching the blanks: distributional similarity for relation learning. arXiv preprint arXiv:1906.03158 (2019)