Coreference Aware Representation Learning for Neural Named Entity Recognition

Zeyu Dai, **Hongliang Fei** and **Ping Li**
Cognitive Computing Lab, Baidu Research USA
jzdaizeyu@gmail.com, hongliangfei@baidu.com, pingli98@gmail.com

Abstract

Recent neural network models have achieved the state-of-the-art performance on the task of named entity recognition (NER). However, previous neural network models typically treat the input sentences as a linear sequence of words but ignore rich structural information, such as the coreference relations among non-adjacent words, phrases or entities. In this paper, we propose a novel approach to learn coreference-aware word representations for the NER task at the document level. In particular, we enrich the well-known neural architecture "CNN-BiLSTM-CRF" with a coreference layer on top of the BiLSTM layer to incorporate coreferential relations. Furthermore, we introduce the coreference regularization to ensure the coreferential entities to share similar representations and consistent predictions within the same coreference cluster. Our proposed model achieves new state-of-the-art performance on two NER benchmarks: CoNLL-2003 and OntoNotes v5.0. More importantly, we demonstrate that our framework does not rely on gold coreference knowledge, and can still work well even when the coreferential relations are generated by a third-party toolkit.

1 Introduction

Named entity recognition (NER) is one of the fundamental tasks in natural language processing (NLP), which has a huge impact on many downstream applications including relation extraction [Li and Ji, 2014], knowledge base completion [Dong *et al.*, 2014] and entity linking [Luo *et al.*, 2015]. Given an input text, NER aims to locate and classify named entities from raw text into pre-defined semantic types such as persons (PER), organizations (ORG), locations (LOC), etc.

The traditional approach for the NER is to regard it as a sequence labeling task, in which each word is assigned with a tag (e.g., "B-PER", "I-PER", "O" in BIO tagging schema) indicating whether the word belongs to part of any named entity or not. To improve the performance of NER, recent NLP researchers usually applied the latest and sophisticated neural sequence labeling models, such as the BiLSTM-CRF [Huang *et al.*, 2015] which first uses the bidirectional LSTMs to process input sentences and then employs the Conditional Random Field (CRF) to label each word jointly.

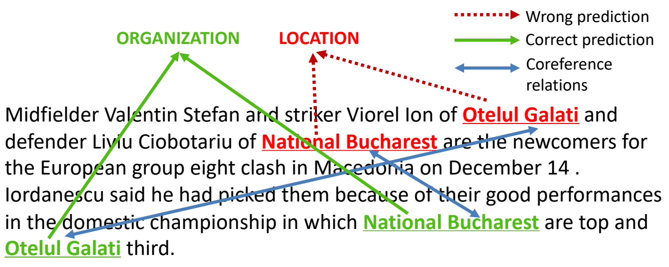
ORGANIZATION
LOCATION
Wrong prediction
Correct prediction
Coreference relations
Midfielder Valentin Stefan and striker Viorel Ion of Otelul Galati and defender Liviu Ciobotariu of National Bucharest are the newcomers for the European group eight clash in Macedonia on December 14 . Iordanescu said he had picked them because of their good performances in the domestic championship in which National Bucharest are top and Otelul Galati third.

Figure 1: Example of inconsistent errors made by Ma and Hovy (2016). **Otelul Galati** and **National Bucharest** are both organization names, but the model of Ma and Hovy (2016) wrongly predicted them as location in the first sentence. Fixing such errors requires incorporating coreference relations into the NER models.

Although recent neural network models have advanced the state-of-the-art performance of NER [Lample *et al.*, 2016; Ma and Hovy, 2016; Peters *et al.*, 2017; Peters *et al.*, 2018], they simply treat the input text as a linear sequence of words but disregard non-sequential structural information such as coreferential relations (i.e., two or more mentions refer to the same person or thing.) between entities whose position can be far away in the raw context. Such a limitation may cause these models to produce globally inconsistent semantic type predictions. Figure 1 shows a typical failure case when applying the well-known model of Ma and Hovy (2016) to two sentences from the CoNLL-2003 dataset [Sang and De Meulder, 2003]. Based on our error analysis, 20%-25% errors made by Ma and Hovy (2016) belong to this category, which can be mostly avoided by utilizing coreference relations. Conceptually, if two entities belong to the same coreference cluster, they should have the same semantic type as well.

Meanwhile, recent end-to-end neural network models for coreference resolution [Lee *et al.*, 2018; Fei *et al.*, 2019] have achieved increasing accuracy over the years which make it possible and practical for us to automatically extract coreferential relations from the raw text without relying on human-annotated coreference knowledge. Therefore, the next key research question is how to incorporate extracted coreference relations into the NER models for predicting consistent semantic type across coreferential entity mentions.

To address the above question, we propose a coreference-aware representation learning framework based on the "CNN-BiLSTM-CRF" NER model [Ma and Hovy, 2016]. Specifically, we design a coreference layer added on top of the BiLSTM layer to incorporate prior knowledge about coreferential relations among entity mentions, either from the ground truth or the external coreference resolvers [Lee *et al.*, 2018]. Besides, we introduce a coreference regularization term to enforce the coreferential words/entities to have similar representations for the NER tag labeling. The combined objective function maximizes both the probability of decoded tag sequence given the input text and the consensus among the coreferential entities' hidden representations.

Our major contributions of this paper can be summarized as follows:

- We present a coreference-aware NER model at the document-level that can explicitly leverage the global structural information of coreferential relations. By introducing a coreference layer and coreference regularization into the base model, our full model enjoys both the strong generalization performance of deep neural network models and the enhancement from coreference guidance. To the best of our knowledge, this is the first neural NER model that effectively exploits the coreference relations.
- We evaluate our model on two benchmarks: OntoNotes v5.0 with gold coreference relations and CoNLL-2003 without gold coreference annotation. On both benchmarks, our full model outperforms all previous approaches with 0.4-1.0% absolute improvement, even when a third-party CoreNLP toolkit generates the coreferential relations.
- Although we focus on improving the NER by using the coreference knowledge in this paper, our model can shed light on other NLP tasks in which there is external knowledge base of structural information available. For example, the entity/event relations can also be incorporated into neural networks for boosting the performance of challenging NLP tasks, including discourse parsing, question answering and natural language understanding.

2 Related Work

2.1 Neural Named Entity Recognition (NER)

Recently, neural network based NER models [Ma and Hovy, 2016; Lample *et al.*, 2016; Strubell *et al.*, 2017] have achieved great improvement over the earlier features-based models. Those neural networks use different strategies (e.g., CNNs or RNNs) to encode characters and words into hidden representations and decode them to named entity tags with a CRF (or LSTM) layer. Another trend for better NER performance is to improve the word embedding with hidden representation depending on word's context by pre-training deep language model on characters or external unlabeled sentences [Liu *et al.*, 2018; Peters *et al.*, 2018; Devlin *et al.*, 2019]. However, most previous works mainly consider NER as a traditional sequence labeling problem but ignore the rich structural information within contexts, such as the coreferential relations used in this paper.

Utilizing external knowledge to improve the NER has also received a lot of attention from the NLP researchers. Jie *et al.* (2017) and Li *et al.* (2017) used the dependency or constituency parse trees to guide NER, while Yang and Mitchell (2017) leveraged external knowledge base to facilitate entity extraction. Durrett and Klein (2014) and Singh *et al.* (2013) proposed features-based joint models to conduct the NER and coreference inference simultaneously. In contrast, our model is an end-to-end deep neural network model and we focus on using the coreferential relations as prior knowledge to learn coreference-aware representations for the NER instead of jointly modeling NER and coreference resolution.

More recently, Luan *et al.* (2018) proposed a deep multi-task framework to jointly perform the NER, relation extraction and coreference resolution, wherein the three tasks shared common hidden layers. However, it suffers from high computational complexity $O(L^4)$, where L is the input sequence length. In addition, the implicit knowledge sharing at the hidden layers cannot explicitly transfer coreference information into the NER task, so it is not as effective as our approach which can explicitly utilize the coreference relations.

2.2 Incorporating Coreference Knowledge into the Neural Network Models

As one important type of linguistic structural information, the coreference knowledge has been explored to improve the performance of neural network models for many NLP applications, such as reading comprehension [Dhingra *et al.*, 2018; Swayamdipta *et al.*, 2018] and relation extraction [Peng *et al.*, 2017]. Since we are the first to exploit coreference knowledge in the neural NER model, we aim to study a task-specific approach for encoding the coreferential relations.

One way to encode the coreferential relations within neural networks is to use external gate (or memory) in RNNs (or memory networks) as the *bottom-level* component [Dhingra *et al.*, 2017; Dhingra *et al.*, 2018]. However, in our point of view, this method can only implicitly utilize the coreference knowledge since the coreference information can be easily lost during the bottom-level forward propagation when processing a long sequence of inputs with increased size of the hidden units (e.g., the model used more hidden units to keep track of each coreference cluster). Another way to explicitly introduce the coreference knowledge is to build the coreference-aware word representations at the *top-level* of neural network models, which usually uses vector transformation functions, including the feedforward neural network, neural tensor network [Socher *et al.*, 2013], soft-attention mechanism [Bahdanau *et al.*, 2015] and others, on top of the word-level RNNs (or CNNs) to refine entity mention representations [Swayamdipta *et al.*, 2018]. In this paper, we follow the top-level approach to design our coreference layer, so that the encoded coreference relations can directly and explicitly influence the final word representations.

3 Our Neural Network Model for NER

In this section, we first introduce the base model for neural named entity recognition, then present the coreference layer which extends the base model by incorporating coreference

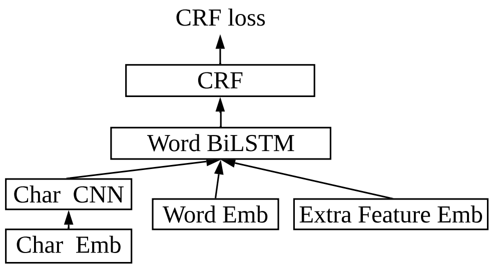

Figure 2: Base Model architecture of CNN-BiLSTM-CRF proposed by Ma and Hovy (2016).

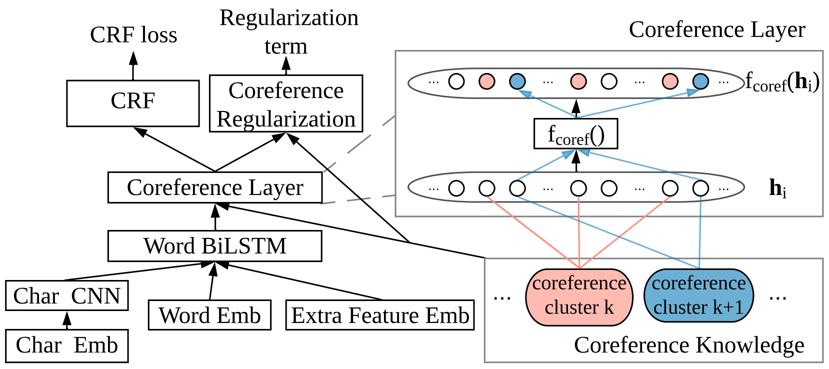

Figure 3: Our document-level model architecture with Coreference Layer and Regularization. The colored arrows and neurons show how to build the coreference-aware word vectors by considering other words within each coreference cluster.

knowledge, and finally propose the coreference regularization which can be used to guide the coreference-aware word representation learning for consistent label prediction of NER.

3.1 Base Model (CNN-BiLSTM-CRF)

In this paper, we choose the CNN-BiLSTM-CRF model [Ma and Hovy, 2016] as our base model since it is the most successful NER model as extensively studied in Yang and Zhang (2018) by comparing different variants of recent neural NER model architectures. As shown in Figure 2, the base model consists of three components, including the character-level CNN, the word-level bidirectional LSTMs and a CRF layer that jointly decodes semantic named entity tags.

Character-level CNN. The character-level features, such as the prefix or suffix of a word, are helpful for alleviating the out-of-vocabulary problem and improving the word representation in neural network models [Dos Santos and Zadrozny, 2014]. In our base model, we adopt one CNN layer with max-pooling operation to extract character-level features $\boldsymbol{w}_i^{char}$ for the i-th word of the input word sequence.

Word-level BiLSTM. Given a word sequence $X = (x_1, x_2, ..., x_L)$ as the input, for each word x_i, we construct an expanded word vector by concatenating its word embedding $\boldsymbol{w}_i^{word}$ with its character-level features and extra word-level features (POS tag) as $\boldsymbol{w}_i = [\boldsymbol{w}_i^{word}, \boldsymbol{w}_i^{char}, \boldsymbol{w}_i^{features}]$. The word-level BiLSTM layer will process the sequence of expanded word vectors $(\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_L)$ by using two separate LSTMs, with one processing the sequence from left to right and the other processing the sequence from right to left. Therefore, at each word index i, we compute two hidden states $\overrightarrow{\boldsymbol{h}}_i, \overleftarrow{\boldsymbol{h}}_i$ and concatenate them to get the word x_i's hidden representation $\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}}_i, \overleftarrow{\boldsymbol{h}}_i]$.

CRF for Sequence Tagging

For the NER task, it is crucial to model the label dependencies (e.g., "I-ORG" must follow "B-ORG" in BIOES [Ratinov and Roth, 2009] tagging schema.) and jointly decode the best label sequence. Therefore, the CRF layer [Collobert *et al.*, 2011] is a better choice for the inference layer since it can dynamically decode a chain of labels and capture the inter-dependency between adjacent labels by maintaining a state-transition matrix as its parameters.

Given the hidden word representations from the BiLSTM $\boldsymbol{H}^{(j)} = (\boldsymbol{h}_1^{(j)}, \boldsymbol{h}_2^{(j)}, ..., \boldsymbol{h}_L^{(j)})$ and the target label sequence $\boldsymbol{y}^{(j)} = (y_1^{(j)}, y_2^{(j)}, ..., y_L^{(j)})$ for the j-th training instance, we minimize the following CRF loss:

$$L_{CRF} = -\sum_j \log p(\boldsymbol{y}^{(j)} | \boldsymbol{H}^{(j)})$$

Here, the conditional probability $p(\boldsymbol{y}|\boldsymbol{H})$ has following form:

$$p(\boldsymbol{y}|\boldsymbol{H}) = \frac{\prod_{i=1}^{L} \psi_i(y_{i-1}, y_i, \boldsymbol{h}_i)}{\sum_{\boldsymbol{y}' \in \mathbb{Y}} \prod_{i=1}^{L} \psi_i(y'_{i-1}, y'_i, \boldsymbol{h}_i)}$$

where $\mathbb{Y}$ denotes the set of all possible label sequences. $\psi_i(y_{i-1}, y_i, \boldsymbol{h}_i) = \exp(\boldsymbol{W}_{y_{i-1},y_i}^T \boldsymbol{h}_i + \boldsymbol{b}_{y_{i-1},y_i})$ is the potential function, in which $\boldsymbol{W}$ and $\boldsymbol{b}$ are trainable parameters. During testing, we use the Viterbi algorithm to search for the optimal label sequence $\boldsymbol{y}^*$ that maximizes the conditional probability: $\boldsymbol{y}^* = \text{argmax}_{\boldsymbol{y} \in \mathbb{Y}} p(\boldsymbol{y}|\boldsymbol{H})$.

3.2 Coreference Layer

As illustrated and discussed in Figure 1, since the base model ignores the coreferential relations, it will likely predict inconsistent named entity tags for coreferential entities. To alleviate this problem, we add a coreference layer between the word-level BiLSTM and the CRF layer into the base model, as shown in Figure 3, to incorporate coreferential relations for learning the coreference-aware word representations.

In this paper, we assume document-level coreference relations are given in the form of coreference clusters, which are either ground truth or generated by a third-party coreference resolver. In practice, there is no overlap between any two coreference clusters, because they will be merged into one cluster if they share any entity mentions. As the input of the coreference layer, let $\mathbb{C} = (\mathbb{C}_1, \mathbb{C}_2, ..., \mathbb{C}_K)$ denotes the coreference clusters in one document, where $\mathbb{C}_k$ contains the

Dataset	Train	Dev	Test	Named Entity types	Document genre	Coreference relation source
CoNLL-2003	23,499	5,942	5,648	4	News	CoreNLP-generated
OntoNotes v5.0	81,828	11,066	11,257	18	News, Web, Mag	gold or CoreNLP-generated

Table 1: Dataset statistics counted in the number of named entities.

word indices with the corresponding words in one document referring to the same entity or thing.[1] For each coreference cluster, we use a feedforward neural network $f_{coref}(.)$ to refine the hidden word representation $\boldsymbol{h}_i$ of coreferential words.

Specifically, given the coreference clusters and hidden word representations from the BiLSTM layer, the output vector of the coreference layer has the following form:

$$f_{coref}(\boldsymbol{h}_i) = \begin{cases} \tanh(\boldsymbol{W}_{coref}[\boldsymbol{h}_i, \boldsymbol{h}_{\mathbb{C}_k}] + \boldsymbol{b}_{coref}), & \text{if } i \in \mathbb{C}_k \\ \boldsymbol{h}_i, & otherwise \end{cases} \quad (1)$$

where $\boldsymbol{W}_{coref}$ and $\boldsymbol{b}_{coref}$ are the weight and bias parameters, $\boldsymbol{h}_{\mathbb{C}_k}$ is the coreference vector calculated by applying max-pooling to all word representations in one cluster:

$$\boldsymbol{h}_{\mathbb{C}_k} = \max_{j \in \mathbb{C}_k} \boldsymbol{h}_j$$

The role of coreference vector is similar to the "context vector" utilized in the soft-attention mechanism [Bahdanau *et al.*, 2015], but we use the simple max-pooling instead of computing weights for different word vectors. Clearly, the output word vector from our coreference layer is influenced by other hidden word representations within the same coreference cluster through the coreference vector.

There is one extreme variant of the coreference layer, which is to directly use the coreference vector as the output with the following form:

$$f_{coref}(\boldsymbol{h}_i) = \begin{cases} \boldsymbol{h}_{\mathbb{C}_k}, & \text{if } i \in \mathbb{C}_k \\ \boldsymbol{h}_i, & otherwise \end{cases} \quad (2)$$

In this variant, all the words within one coreference cluster share the same representation for label tagging. We will compare and discuss these variants in the following Section 5.2.

3.3 Coreference Regularization

Conceptually, the hidden word representations within one coreference cluster should be similar, so that the CRF layer can make consistent predictions across different coreferential mentions. To guide the word representation learning of the coreference layer, we propose two types of regularization and apply them to the output word vectors of above coreference layer. The resulting regularization term is also minimized as a part of the final objective function during model training.

The first one is "Euclidean Coreference Regularization", which calculates the Euclidean distance to penalize the difference between two coreferential word vectors. The coreference regularization term has the following form:

$$R_{coref} = \sum_{k} \sum_{(i,j) \in \mathbb{C}_k} ||f_{coref}(\boldsymbol{h}_i) - f_{coref}(\boldsymbol{h}_j)||_2$$

[1]If one entity mention has multiple words (e.g., Los Angeles), we only keep the first word's index because it is much easier to decode the rest words' tags for the CRF layer if the first tag is correct.

The second one is "Cosine Coreference Regularization", which uses the cosine similarity to measure two word vectors' similarity. The coreference regularization term is as follows:

$$R_{coref} = \sum_{k} \sum_{(i,j) \in \mathbb{C}_k} (1 - \cos(f_{coref}(\boldsymbol{h}_i), f_{coref}(\boldsymbol{h}_j)))$$

Hence the overall objective function for our full model is:

$$L = L_{CRF} + \lambda \times R_{coref}$$

In Section 5.2, we will compare two types of coreference regularization we proposed and discuss the reasonable strategy to set the coreference regularization parameter λ.

4 Experiments

4.1 Datasets

We evaluated our model on two standard NER datasets including CoNLL-2003 [Sang and De Meulder, 2003] and OntoNotes v5.0 [Pradhan *et al.*, 2011]. Table 1 gives an overview and statistics of the two datasets. The CoNLL-2003 was annotated with four coarse-grained entity types, including Person (PER), Location (LOC), Organization (ORG) and Miscellaneous (MISC), while the OntoNotes was annotated with 18 fine-grained named entity types. Compared with CoNLL-2003, the OntoNotes corpus was much larger and covered a wider variety of text genres including broadcast news, new testaments, magazine and Web text. Following previous work [Chiu and Nichols, 2016; Ghaddar and Langlais, 2018], we excluded the new testaments portion of OntoNotes from both train and test sets since this portion did not provide gold annotation of entity tags. Since coreference relations were not annotated on the CoNLL-2003 dataset, we utilized the latest version of Stanford CoreNLP toolkit [Manning *et al.*, 2014] to extract coreference clusters in each document, although this toolkit can only achieve around 60% F1-score for coreference resolution on the CoNLL-2011/2012 dataset. To be directly comparable with previous work, we used the official train/dev/test set splits on both datasets.

4.2 Experiment Setting

Preprocess. Following previous work [Ma and Hovy, 2016], we replaced all digit characters with '0' and converted the tagging schema from BIO to BIOES which additionally used "E-" and "S-" to represent the end of entity and single-word entity respectively. Following suggestions of Yang and Zhang (2018), we used part-of-speech (POS) tag and capitalization (Cap) flag of each word as extra features, which will be concatenated with word embedding. For all coreference clusters, we masked the word index of personal pronouns (e.g., him, she, it) because these words are not named

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
char emb size	30	word emb size	100	POS emb	15	optimizer	SGD with momentum 0.9
char CNN units	50	word LSTM units	300	Cap emb	5	initial lr	0.015
char CNN window	3	word LSTM layer	1	dropout	0.5	batch size	1 (CoNLL)/ 5 (OntoNotes)
char CNN layer	1	ELMo emb size	1024	lr decay	5%	coref λ	0.01,0.1,0.5,1.0,1.5,2.0,5.0

Table 2: Hyperparameters of the model used in our experiments.

entities and won't help identify named entities within same coreference cluster.[2]

Fixed vs. Dynamic Word Embedding. Pre-trained word embedding such as GloVe had a limitation that each word's representation is fixed without considering its context, which conflicts with the fact one word can have different meanings in different contexts. Recent work, including AllenAI's ELMo [Peters *et al.*, 2018] and Google's BERT [Devlin *et al.*, 2019], showed that context-dependent (dynamic) word representations learned from deep language model can benefit neural networks for challenging NLP tasks, which outperformed the traditional fixed word embedding. In this work, we tried both GloVe word embedding (100D) and dynamic ELMo[3] embedding (1024D) to initialize our word embedding.

Hyperparameters. Table 2 summarizes the hyperparameters used in our experiments, which mostly followed Yang *et al.* (2018), including using the SGD optimizer with a decayed learning rate to update parameters. Since the OntoNotes corpus was much larger than the CoNLL-2003 dataset, we used a larger batch size to speed up the model training. For the coreference regularization parameter λ, we tuned it based on the best performance on the dev set. To prevent gradient exploding, we clipped the gradient L2-norm with a threshold of 5.0 and used the L2 regularization with coefficient 10^{-8}.

Evaluation. We adopt the standard entity-level micro-averaged F1-score as the main evaluation metrics. To diminish the effects of randomness in training neural network models and report stable experimental results, we ran all our proposed model, its variants as well as the base model 10 times and reported the averaged F1-score and standard deviation over multiple trials. For a fair comparison, all our models were implemented with Pytorch and evaluated on a Nvidia Titan X GPU using the same random seed.

4.3 Experiment Results

Table 3 shows the performance of our proposed models compared to the recent published models on the CoNLL-2003 test set. The first section of the table lists the models which did not use any external data other than train set and pre-trained fixed word embedding, while the models in the second section utilized external data for different purposes (e.g., training language model for word embedding [Peters *et al.*, 2018] or doing transfer learning [Yang *et al.*, 2017]). In the third section, our replicated CNN-BiLSTM-CRF model is slightly worse than the one initially reported in Ma and Hovy (2016). One possible reason is that we conduct document-level NER tagging rather than original sentence-level experiments (we obtain 91.24% F1-score for the sentence-level NER tagging).

Model	F1 Score (± std)	
	Train	Train + Dev
Previous work w/o using external data		
[Huang *et al.*, 2015]	90.10	-
[Strubell *et al.*, 2017]	90.65 (± 0.15)	-
[Shen *et al.*, 2018]	90.89 (± 0.19)	-
[Lample *et al.*, 2016]	90.94	-
[Ma and Hovy, 2016]	91.21	-
[Liu *et al.*, 2018]*	91.24 (± 0.12)	-
[Ye and Ling, 2018]*	91.38 (± 0.10)	-
Previous work w/ using external data		
[Chiu and Nichols, 2016]	91.23 (± 0.16)	91.62 (± 0.33)
[Yang *et al.*, 2017]	91.26	-
[Tran *et al.*, 2017]*	91.69	-
[Peters *et al.*, 2017]*	-	91.93 (± 0.19)
ELMo [2018]*	-	92.22 (± 0.10)
BERT Base [2019]*	92.40	-
CVT + MultiTask [2018]*	92.61 (± 0.09)	-
Document-level NER in this work		
CNN-BiLSTM-CRF	90.88 (± 0.13)	91.31 (± 0.13)
+ coreference layer	91.53 (± 0.14)	91.86 (± 0.11)
+ coref regularization	91.65 (± 0.15)	92.03 (± 0.14)
+ ELMo embedding*	**93.19** (± 0.13)	**93.37** (± 0.14)

Table 3: NER performance on the test set of CoNLL-2003. "Train + Dev" indicates that both the train and dev sets were used for model training after tuning hyperparameters on the dev set. The models marked with * utilized word embedding from deep language model.

Built on top of the base model, the coreference layer improves the NER performance by 0.65 points on average (statistical significant t-test with $p < 0.01$). Using the coreference regularization to guide the coreference-aware word representation learning can improve the result (statistical significant t-test with $p < 0.05$ comparing to not using the coreference regularization), but by a small margin. As shown in the last column, introducing the context-dependent ELMo embedding boosts the performance of NER, which further validates our model's utility when combining with the latest word embedding techniques. Noticing that our full model significantly outperforms the ELMo baseline (the fifth row in the second section) by (93.37 - 92.22 = 1.15) points, we conclude that our approach with coreference layer and coreference regularization can effectively improve the NER performance and plays a key role in achieving the best performance.

Overall, our full model achieves the state-of-the-art performance of 93.19% F1-score when using the dynamic word embedding from language model (comparing to models marked

[2]We also tried not to mask personal pronouns, and the resulting F1 score slightly decreased by 0.08 on the CoNLL-2003 dataset.

[3]From the AllenAI's website (https://allennlp.org/elmo), we downloaded the pre-trained ELMo embedding trained on 5.5B tokens and froze its parameters during our NER model training.

Model	F1 Score ($\pm$ std)
Previous work	
[Chiu and Nichols, 2016]	86.41 ($\pm$ 0.22)
[Shen *et al.*, 2018]	86.63 ($\pm$ 0.49)
[Strubell *et al.*, 2017]	86.99 ($\pm$ 0.22)
[Li *et al.*, 2017]	87.21 ($\pm$ 0.14)
[Ghaddar and Langlais, 2018]	87.95 ($\pm$ 0.13)
CVT + MultiTask [2018]	88.81 ($\pm$ 0.09)
JointNERCoref [Luan *et al.*, 2018][4]	84.09 ($\pm$ 0.16)
Document-level NER in this work	
CNN-BiLSTM-CRF	88.06 ($\pm$ 0.13)
Use CoreNLP-generated coreference knowledge	
+ coreference layer	88.67 ($\pm$ 0.15)
+ coref regularization	88.95 ($\pm$ 0.13)
+ ELMo embedding	**89.83** ($\pm$ 0.11)
Use gold coreference knowledge	
+ coreference layer	88.85 ($\pm$ 0.13)
+ coref regularization	89.08 ($\pm$ 0.12)
+ ELMo embedding	89.96 ($\pm$ 0.10)

Table 4: NER performance on the test set of OntoNotes v5.0. Note that CVT + MultiTask [Clark *et al.*, 2018] used the ELMo word embedding, so this result is only comparable with our last row.

Model	F1 Score	
	Mean ($\pm$ std)	Max
Full Model	93.19 ($\pm$ 0.13)	93.35
w/o coref regularization	92.86 ($\pm$ 0.11)	93.04
w/o coref layer & regularization	92.32 ($\pm$ 0.10)	92.45
w/o GloVe embedding	93.03 ($\pm$ 0.11)	93.14
w/o Character-level CNN	93.13 ($\pm$ 0.10)	93.27
w/o ELMo embedding	91.62 ($\pm$ 0.21)	91.96

Table 5: Ablation study on the CoNLL-2003. We report the NER performance when each component was removed from the full model. We set $\lambda = 1.0$ if the coreference regularization was used.

with * in Table 3), and our approach simultaneously obtains the best F1-score of 91.65% when using the fixed word embedding *only*. This result proves our model can effectively work without relying on the gold coreferential relations.

Table 4 reports our experimental results as well as previous approaches that were evaluated on the OntoNotes v5.0 test set. Similar to the result on the CoNLL-2003, better NER performance can be achieved by using our coreference layer and coreference regularization. Therefore, our full NER model achieves the state-of-the-art F1-score of 89.83% on the OntoNotes as well, which outperforms the previous best-published result (88.81%) of Clark *et al.* (2018) by a large margin. Again, we demonstrate that the noisy coreference knowledge extracted from an external system is still usable in our framework for improving the NER performance.

It is worth mentioning that the latest multi-task joint NER and coreference resolution model [Luan *et al.*, 2018] performs worse than our method, even worse than those traditional sequence labeling approaches. One possible reason is

[4]Code was downloaded from https://bitbucket.org/luanyi/scierc/src/master/. We set $rel\_weight = 0$ to disable the relation inference task and carefully tuned other network architecture hyperparameters over a wide range on the validation set.

Dataset	CoNLL-2003		OntoNotes v5.0	
λ	Mean ($\pm$ std)	Max	Mean ($\pm$ std)	Max
0.01	91.48 ($\pm$ 0.19)	91.83	89.03 ($\pm$ 0.08)	89.18
0.1	**91.65** ($\pm$ 0.15)	91.89	89.07 ($\pm$ 0.17)	**89.32**
0.5	91.60 ($\pm$ 0.11)	91.74	**89.08** ($\pm$ 0.12)	89.22
1.0	91.62 ($\pm$ 0.21)	**91.96**	88.89 ($\pm$ 0.10)	89.09
1.5	91.53 ($\pm$ 0.11)	91.71	-	-
2.0	91.46 ($\pm$ 0.12)	91.66	-	-
5.0	91.44 ($\pm$ 0.11)	91.65	-	-

Table 6: Impact of the coreference regularization λ on the Base Model + Coreference Layer + Coreference Regularization.

that the implicit knowledge transfer by sharing intermediate layers for the NER and coreference resolution tasks is not as effective as explicitly imposing the coreference regularization. Besides, brute force enumeration of all possible span candidates ignores the inner dependency among sub-chunks of entities, which can be better captured by sequence labeling.

5 Analysis and Discussion

5.1 Ablation Study

To understand how much contribution each major component of our full model makes, we did an ablation study as illustrated in Table 5. Clearly, removing the coreference regularization and coreference layer significantly deteriorates the model performance by 0.3 and 0.9 points respectively (statistical significant t-test with $p < 0.01$). Among three parts of expanded word vector including GloVe, characters and ELMo, the ELMo embedding makes the most contribution and improves the NER result by 1.5 points on average which is consistent with the observation of Peters *et al.* (2018).

5.2 Choice of Coreference Layer and Coreference Regularization

Similar vs. Same Coreferential Representations. On the CoNLL-2003, we evaluated two variants of the coreference layer introduced in Section 3.2. The comparative study under the same settings (base model + coreference layer w/o coref regularization and ELMo) showed that, as the input of the CRF layer, similar coreferential representations (91.53% F1-score) generated by Equation (1) are significantly better than the exact same coreferential representations (91.21% F1-score) as the output of Equation (2). One potential explanation is that the important context information of individual entity is missing if we enforce all entities within one coreference cluster to share the same representation in vector space.

Cosine vs. Euclidean Coreference Regularization. We compared two types of coreference regularization introduced in Section 3.3 on the CoNLL-2003. The experiment result showed that the "Cosine Coreference Regularization" clearly outperformed the "Euclidean Coreference Regularization" with 91.65% vs. 91.23% F1-score (we set $\lambda = 0.1$ for a fair comparison) which meets our expectation that the cosine similarity can better parameterize the similarity between two vectors than the euclidean distance in vector space.

Error	Base Model (CNN-BiLSTM-CRF)	Base Model + Coreference Layer & Regularization
Boundary	[Defender Hassan Abbas]$_{(PER)}$ ... [Hassan Abbas]$_{(PER)}$	Defender **[Hassan Abbas]**$_{(PER)}$ ... *[Hassan Abbas]*$_{(PER)}$
Mixed Entities	[Australia]$_{(LOC)}$ vs. [West Indies World Series]$_{(LOC)}$ ... Scoreboard in the [World Series]$_{(MISC)}$ limited overs match between [Australia]$_{(LOC)}$ and [West Indies]$_{(LOC)}$	Australia vs. **[West Indies]**$_{(LOC)}$ **[World Series]**$_{(MISC)}$ ... Scoreboard in the *[World Series]*$_{(MISC)}$ limited overs match between [Australia]$_{(LOC)}$ and *[West Indies]*$_{(LOC)}$
Inconsistent Type	[Liviu Ciobotariu]$_{(PER)}$ of [National Bucharest]$_{(LOC)}$ ... championship in which [National Bucharest]$_{(ORG)}$	[Liviu Ciobotariu]$_{(PER)}$ of **[National Bucharest]**$_{(ORG)}$ ... championship in which *[National Bucharest]*$_{(ORG)}$
Inconsistent Boundary	Arab win [African Cup]$_{(MISC)}$ [Winners' Cup]$_{(MISC)}$ ... Result of the [African Cup Winners' Cup]$_{(MISC)}$ final	Arab win **[African Cup Winners' Cup]**$_{(MISC)}$ ... Result of the *[African Cup Winners' Cup]*$_{(MISC)}$ final
Similar Entities	[Melbourne Cricket Ground]$_{(ORG)}$ ... [Sydney Cricket Ground]$_{(LOC)}$ ... [Melbourne]$_{(LOC)}$ Cricket Ground	**[Melbourne Cricket Ground]**$_{(LOC)}$ ... *[Sydney Cricket Ground]*$_{(LOC)}$ ... **[Melbourne Cricket Ground]**$_{(LOC)}$

Table 7: Case study on the CoNLL-2003 dataset. We summarize typical types of errors fixed by using our coreference layer and coreference regularization *w/o ELMo*. Named entities in red (underling) text are wrongly predicted with labels in the round brackets. Named entities in green (bold) text are correctly predicted by considering coreferential entities in blue (italic) text using our coreference-aware approach.

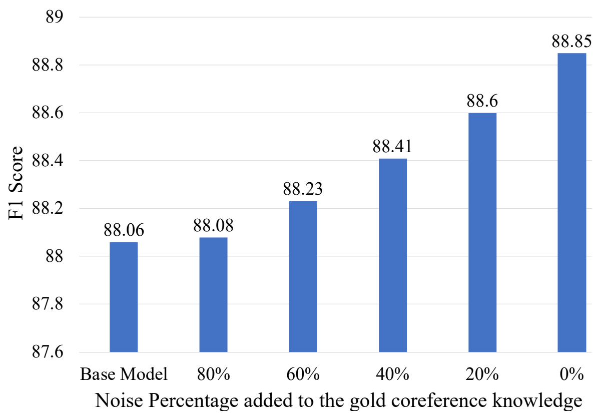

Figure 4: Impact of the coreference knowledge quality on the Base Model + Coreference Layer (on the OntoNotes v5.0).

Impact of Coreference Regularization λ. From Table 6, we can see that the coreference regularization parameter λ has a nontrivial effect on the performance of our model. We recommend to choose λ from the range [0.1, 1.0] and tune it based on the density of coreference relations in the data (e.g., smaller λ for higher coreference relations density).

5.3 Impact of Coreference Knowledge Quality

Since gold coreference knowledge is rare and valuable, it is important for our framework to tolerate the noisy coreferential relations as prior knowledge. In order to study the influence of the coreference quality on our model, we gradually add noise into the OntoNotes' gold coreference clusters by randomly deleting or fluctuating (± 5) the coreferential entities' indices with a certain probability (i.e., noise percentage). As shown in Figure 4, the F1 score increases quickly with less noise and our coreference layer can still improve the NER performance with 60% noise in the coreference knowledge which demonstrates the robustness of our model.

5.4 Case Study

To study the behavior of our proposed model and better understand what types of errors made by Ma and Hovy (2016) were corrected by our coreference-aware approach, we did an error analysis on the CoNLL-2003 dataset and listed a few representative examples in Table 7. To make our contribution clear, we did not use the ELMo embedding for both models.

As shown in the Table 7, our approach not only helps correctly predict the semantic type of coreferential entities within a coreference cluster (the second, third and fifth example), but also locates the accurate boundary of coreferential named entities (the first, fourth and fifth example).

6 Conclusion

We present a novel neural network model for the NER task which builds the coreference-aware word representations by explicitly utilizing the coreferential relations with our proposed coreference layer. Furthermore, we introduce the coreference regularization to ensure the coreferential entities to share similar representations and consistent predictions within the same coreference cluster. Experiments on two benchmarks demonstrate that our full model with the coreference layer and coreference regularization significantly outperforms all previous NER systems, even given the noisy coreference information as prior knowledge.

References

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[Chiu and Nichols, 2016] Jason Chiu and Eric Nichols. Named entity recognition with bidirectional LSTM-CNNs. *TACL*, 4:357–370, 2016.

[Clark *et al.*, 2018] Kevin Clark, Minh-Thang Luong, Christopher Manning, and Quoc Le. Semi-supervised sequence modeling with cross-view training. In *EMNLP*, pages 1914–1925, 2018.

[Collobert *et al.*, 2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[Dhingra *et al.*, 2017] Bhuwan Dhingra, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. Linguistic knowledge as memory for recurrent neural networks. *arXiv preprint arXiv:1703.02620*, 2017.

[Dhingra *et al.*, 2018] Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. Neural models for reasoning over multiple mentions using coreference. In *NAACL-HLT*, pages 42–48, 2018.

[Dong *et al.*, 2014] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *ACM SIGKDD*, pages 601–610, 2014.

[Dos Santos and Zadrozny, 2014] Cicero Nogueira Dos Santos and Bianca Zadrozny. Learning character-level representations for part-of-speech tagging. In *ICML*, pages 1818–1826, 2014.

[Durrett and Klein, 2014] Greg Durrett and Dan Klein. A joint model for entity analysis: Coreference, typing, and linking. *TACL*, 2:477–490, 2014.

[Fei *et al.*, 2019] Hongliang Fei, Xu Li, Dingcheng Li, and Ping Li. End-to-end deep reinforcement learning based coreference resolution. In *ACL*, 2019.

[Ghaddar and Langlais, 2018] Abbas Ghaddar and Philippe Langlais. Robust lexical features for improved neural network named-entity recognition. In *COLING*, pages 1896–1907, 2018.

[Huang *et al.*, 2015] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[Jie *et al.*, 2017] Zhanming Jie, Aldrian Obaja Muis, and Wei Lu. Efficient dependency-guided named entity recognition. In *AAAI*, pages 3457–3465, 2017.

[Lample *et al.*, 2016] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *NAACL-HLT*, pages 260–270, 2016.

[Lee *et al.*, 2018] Kenton Lee, Luheng He, and Luke Zettlemoyer. Higher-order coreference resolution with coarse-to-fine inference. In *NAACL-HLT*, pages 687–692, 2018.

[Li and Ji, 2014] Qi Li and Heng Ji. Incremental joint extraction of entity mentions and relations. In *ACL*, pages 402–412, 2014.

[Li *et al.*, 2017] Peng-Hsuan Li, Ruo-Ping Dong, Yu-Siang Wang, Ju-Chieh Chou, and Wei-Yun Ma. Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks. In *EMNLP*, pages 2664–2669, 2017.

[Liu *et al.*, 2018] Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Fangzheng Xu, Huan Gui, Jian Peng, and Jiawei Han. Empower sequence labeling with task-aware neural language model. In *AAAI*, 2018.

[Luan *et al.*, 2018] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *EMNLP*, pages 3219–3232, 2018.

[Luo *et al.*, 2015] Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. Joint entity recognition and disambiguation. In *EMNLP*, pages 879–888, 2015.

[Ma and Hovy, 2016] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *ACL*, pages 1064–1074, 2016.

[Manning *et al.*, 2014] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL: system demonstrations*, pages 55–60, 2014.

[Peng *et al.*, 2017] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-sentence n-ary relation extraction with graph LSTMs. *TACL*, 5(1):101–115, 2017.

[Peters *et al.*, 2017] Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. In *ACL*, pages 1756–1765, 2017.

[Peters *et al.*, 2018] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL-HLT*, pages 2227–2237, 2018.

[Pradhan *et al.*, 2011] Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *CoNLL: Shared Task*, pages 1–27, 2011.

[Ratinov and Roth, 2009] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the thirteenth conference on computational natural language learning*, pages 147–155. Association for Computational Linguistics, 2009.

[Sang and De Meulder, 2003] Erik Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *NAACL-HLT*, 2003.

[Shen *et al.*, 2018] Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition. In *ICLR*, 2018.

[Singh *et al.*, 2013] Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. Joint inference of entities, relations, and coreference. In *AKBC@CIKM 2013*, 2013.

[Socher *et al.*, 2013] Richard Socher, Danqi Chen, Christopher Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *NIPS*, 2013.

[Strubell *et al.*, 2017] Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. Fast and accurate entity recognition with iterated dilated convolutions. In *EMNLP*, 2017.

[Swayamdipta *et al.*, 2018] Swabha Swayamdipta, Ankur Parikh, and Tom Kwiatkowski. Multi-mention learning for reading comprehension with neural cascades. *ICLR*, 2018.

[Tran *et al.*, 2017] Quan Tran, Andrew MacKinlay, and Antonio Jimeno Yepes. Named entity recognition with stack residual LSTM and trainable bias decoding. In *IJCNLP*, pages 566–575, 2017.

[Yang and Mitchell, 2017] Bishan Yang and Tom Mitchell. Leveraging knowledge bases in LSTMs for improving machine reading. In *ACL*, pages 1436–1446, 2017.

[Yang and Zhang, 2018] Jie Yang and Yue Zhang. Ncrf++: An open-source neural sequence labeling toolkit. In *ACL*, 2018.

[Yang *et al.*, 2017] Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. Transfer learning for sequence tagging with hierarchical recurrent networks. In *ICLR*, 2017.

[Yang *et al.*, 2018] Jie Yang, Shuailong Liang, and Yue Zhang. Design challenges and misconceptions in neural sequence labeling. In *COLING*, 2018.

[Ye and Ling, 2018] Zhixiu Ye and Zhen-Hua Ling. Hybrid semi-markov CRF for neural sequence labeling. In *ACL*, pages 235–240, 2018.