Attention and Edge-Label Guided Graph Convolutional Networks for Named Entity Recognition

Renjie Zhou[1], **Zhongyi Xie**[1], **Jian Wan**[2], **Jilin Zhang**[1], **Yong Liao**[3] and **Qiang Liu**[4]
[1]Hangzhou Dianzi University, [2]Zhejiang University of Science and Technology,
[3]Univerisity of Science and Technology of China, [4]Zhejiang Police College
{rjzhou, zyxie}@hdu.edu.cn

Abstract

It has been shown that named entity recognition (NER) could benefit from incorporating the long-distance structured information captured by dependency trees. However, dependency trees built by tools usually have a certain percentage of errors. Under such circumstances, how to better use relevant structured information while ignoring irrelevant or wrong structured information from the dependency trees to improve NER performance is still a challenging research problem. In this paper, we propose the Attention and Edge-Label guided Graph Convolution Network (AELGCN) model. Then, we integrate it into BiLSTM-CRF to form BiLSTM-AELGCN-CRF model. We design an edge-aware node joint update module and introduce a node-aware edge update module to explore hidden structured information entirely and solve the wrong dependency label information to some extent. After two modules, we apply attention-guided GCN, which automatically learns how to attend to the relevant structured information selectively. We conduct extensive experiments on several standard datasets across four languages and achieve better results than previous approaches. Through experimental analysis, it is found that our proposed model can better exploit the structured information on the dependency tree to improve the recognition of long entities.

1 Introduction

Named Entity Recognition (NER) is the recognition of entities with specific meanings in the text, mainly including person, organization, location, etc. NER is the fundamental tasks for many natural language processing tasks such as relation extraction (Miwa and Bansal, 2016), event extraction (Chen et al., 2015; Liu et al., 2019), coreference resolution (Lee et al., 2017), question answer (Yao and Van Durme, 2014), and knowledge graph (Li et al., 2022). Previous studies have obtained useful structured information from dependency trees and have verified the effectiveness of integrating syntactic dependency into NER tasks (Jie et al., 2017; Jie and Lu, 2019; Aguilar and Solorio, 2019; Xu et al., 2021).

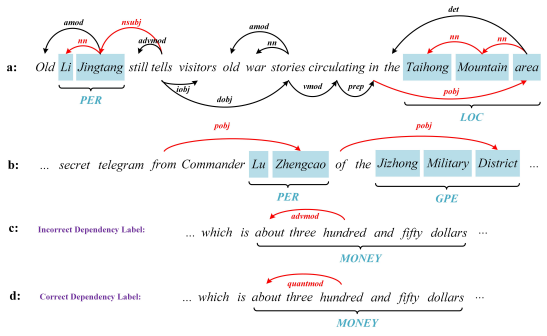

Figure 1: Sentences annotated with dependency tree and named entities.

A dependency tree reveals the syntactic structure of a language unit by analyzing the dependency relationships between its components, and "dependency" refers to the relationship between related words while specifying a dependency relationship to form a syntactic tree reflecting the syntactic relationships between words in a sentence. However, how to better use relevant structured information while ignoring irrelevant or wrong structured information for NER remains a research question to be answered.

For example, Figure 1 (a) shows the dependency tree and named entities of the sentence "Old Li Jingtang still tells visitors old war stories circulating in the Taihong Mountain area". For the PER (person) entity "*Li Jingtang*" where there is an external dependency arc, which is pointed from the word "*tells*" to "*Li Jingtang*", whose dependency label is "*nsubj*" (nominal subject). Similarly, for the LOC (location) entity "*Taihong Mountain area*", which has an incoming arc with dependency label "*pobj*" (prepositional object). These dependency arcs contain structured information between words and dependency label information that is useful

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 6499 - 6510
December 7-11, 2022

for identifying named entity categories and boundaries.

However, under a different context, as shown in Figure 1 (b), the same dependency label (*pobj*) may convey different information for NER. For example, the dependency label "*pobj*" connected with "*area*" indicates LOC entity, but another dependency label "*pobj*" connected with "*Zhengcao*" points out PER entity. Therefore, a single context-independent representation for each dependency label is not enough to express the complex relations between words (Cui and Chen, 2022).

Even worse, there are some incorrect dependency labels in the dataset, these incorrect dependency labels may convey the wrong information for NER. Figure 1 (c,d) shows that for the same MONEY entity "*about three hundred and fifty dollars*", there is an external dependency arc pointed from the word "*hundred*" to "*about*" in sentence (c) and (d), respectively. But the label of dependency in sentence (c) is incorrect. The incorrect dependency label "*advmod*" (adverbial modifier) would convey wrong information, as a result, "*three hundred and fifty dollars*" would probably be recognized as the MONEY entity by mistake. Thus it is necessary to find a solution to mitigate the impact of incorrect dependency labels.

Also, on the other hand, sequence models like bidirectional LSTM (Hochreiter and Schmidhuber, 1997) are not able to fully capture the long-range dependencies (Bengio, 2009). Using the structured information contained in the dependency tree can solve the problem to some extent.

We propose a novel dependency-based named entity recognition model to address the above problems, which improves the named entity recognition performance by exploiting syntactic dependency information with graphical neural networks. The model obtains contextual information by BiLSTM and then by Attention and Edge-Label guided Graph Convolution Network (AELGCN) to integrate better contextual and structured information. For each AELGCN layer, an edge-aware node joint update module is firstly performed for aggregating information from neighbors and different dependency labels. Then a node-aware edge update module is used to update the dependency label representation by its connected node representations, which makes dependency label representation more informative. After that, we introduce attention-guided GCN (AGGCN) (Guo et al., 2019) which contains an attention guide, densely connected, and linear combination layer. The AGGCN is able to select and discard structured information. These two modules with AGGCN are complementary to each other and work in a mutual promotion way. Finally, the Conditional Random Field Model (CRF) (Lafferty et al., 2001b) is used to predict the labels of entities.

Our contributions can be summarized as follows:

- We propose an edge-aware node joint update module and introduce a node-aware edge update module. These two modules exploit the adjacency matrix and dependency label embedding adjacency matrix to learn structured information representation in a context-dependent manner and mitigate the impact of incorrect dependency labels.

- We introduce AGGCN, which exploits the multi-head self-attention mechanism better learn how to select effective structured information. We combine the two modules with AGGCN to construct our proposed AELGCN model. Finally, we integrate AELGCN into the BiLSTM-CRF model to form a novel model called the BiLSTM-AELGCN-CRF model. The model effectively leverages the structured information, thus improving the performance of NER.

- We have conducted extensive experiments on standard datasets across four languages. On these datasets, our proposed model significantly outperforms previous approaches.

2 Related Work

The traditional feature-based NER approaches require considerable feature engineering skills and domain expertise. However, deep neural network based models can build reliable NER systems with much less effort in designing features. BiLSTM-CRF model (Huang et al., 2015) is one of the earliest neural network based models. Due to word embeddings having the out-of-vocabulary problem in BiLSTM-CRF, character-level embeddings generated by LSTM (Lample et al., 2016) or CNN (Ma and Hovy, 2016) are concatenated to enhance the representation of rare and out-of-vocabulary words. To further improve named entity recognition, the representation of words was later enhanced by pre-trained language models (Peters et al., 2018; Devlin

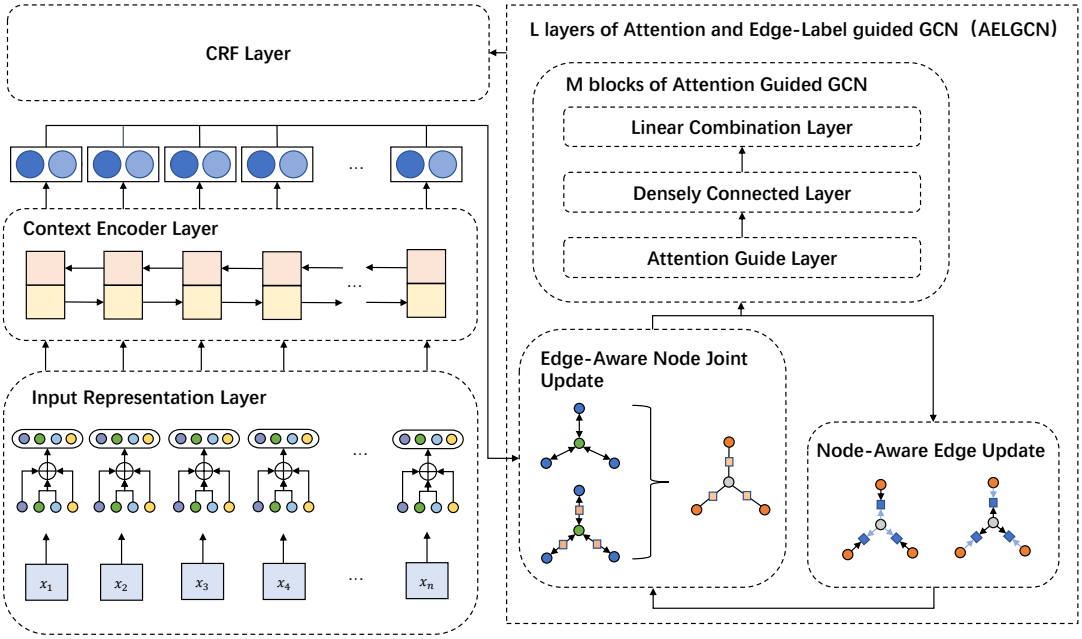

Figure 2: Illustration of BiLSTM-AELGCN-CRF architecture. After input representation and context encoder layer, L layers of AELGCN are stacked to learn syntax-enhanced word representation for sequence labeling. AELGCN is composed of three modules: Edge-Aware Node Joint Update Module, which aggregates information from neighbors of each node and dependency label information, Node-Aware Edge Update Module, which updates context-independent representation for each dependency label, and M blocks of Attention Guided GCN, which learns how to select effective structured information.

et al., 2019). Recent works have focused on leveraging sentence and document-level representations into NER models. (Luo et al., 2020) enhanced the sentence representation, which was learned from an independent BiLSTM via label embeddings, and used key-value memory networks with an attention mechanism to calculate document-level representations. (Schweter and Akbik, 2020) used the contextual information of the current sentence to generate a contextual representation of the sentence so that more information could be obtained for the sentence representation. These models focus on finding good contextualized word representations and better sentence representations to improve NER.

Syntactic information also plays an important role in NER. (Jie et al., 2017) exploited TREE-LSTM (Tai et al., 2015) to extract the dependency tree features and a semi-Markov model to predict the entity types. (Cetoli et al., 2017) found that the use of dependency tree information through graph convolutional networks (GCN) (Kipf and Welling, 2017) has been effective for named entity recognition. (Jie and Lu, 2019) proposed a DGLSTM-CRF model by introducing dependencies into an LSTM to obtain information about the dependency tree for named entity recognition. (Xu et al., 2021) proposed the Synergized-LSTM (Syn-LSTM), where they constructed syntactic adjacency matrices and learned syntactic information through GCN, allowing the LSTM cell to update and represent hidden states with additional GCN syntactic information representations. Although these previous approaches have utilized dependency tree structures, we focus on exploring neural architectures to better exploit structured information by dependency trees.

3 Model

This section presents our BiLSTM-AELGCN-CRF model in detail. Figure 2 shows the overall model architecture, which consists of four components: the Input Representation Layer, the Context Encoder Layer, the Attention and Edge-Label guided GCN layer, and the CRF Layer.

3.1 Input Representation Layer

Following the work by (Xu et al., 2021), given a sequence of n tokens $X = \{x_1, x_2, ..., x_n\}$, for each word x_t, the input representation x_t of our model is the concatenation of the word embedding w_t, the character representation c_t, the dependency

relation embedding r_t, and the POS embedding p_t:

$$x_t = [w_t; c_t; r_t; p_t] \tag{1}$$

where w_t is the pre-trained word embedding, character-level embedding c_t is learned from character-based BiLSTM, r_t and p_t embeddings are randomly initialized and fine-tuned during training. In addition, we use contextualized representations such as BERT (Devlin et al., 2019) in our experiments, we further concatenate the contextual word representation to x_t.

3.2 Context Encoder Layer

Given the input representation x, then x is fed into BiLSTM, which is applied to generate contextual representation. The BiLSTM enables the model to get contextual information from both directions.

$$\begin{aligned} H &= \{h_1, h_2, ..., h_n\} \\ &= BiLSTM(x_1, x_2, ..., x_n) \\ h_t &= [\overrightarrow{h}_t, \overleftarrow{h}_t] \\ \overrightarrow{h_t} &= LSTM(x_t, \overrightarrow{h}_{t-1}, \overrightarrow{\theta}) \\ \overleftarrow{h_t} &= LSTM(x_t, \overleftarrow{h}_{t-1}, \overleftarrow{\theta}) \end{aligned} \tag{2}$$

where $\overrightarrow{\theta}$ and $\overleftarrow{\theta}$ are learnable parameters, respectively.

3.3 Attention and Edge-Label guided Graph Convolutional Networks

In this subsection, we first introduce the GCN model and then present the proposed AELGCN, which contains an edge-aware node joint update module, a node-aware edge update module and the attention-guided GCN.

3.3.1 Vanilla Graph Convolutional Network

GCN (Kipf and Welling, 2017), which is capable of encoding graphs, is an extension of convolutional neural network. For an L-layer GCN, if we denote H^{l-1} the input state and H^l the output state of the l-th layer, the graph convolutional operation can be formulated as:

$$\begin{aligned} H^l &= GCN(A, H^{l-1}, W) \\ &= \sigma(AH^{l-1}W) \end{aligned} \tag{3}$$

where $H^l = \{h_1^l, h_2^l, ..., h_n^l\}$, $l \in [1, 2, ..., L]$, the calculation formula is as follows:

$$h_i^l = \sigma(\sum_{j=1}^{n} A_{ij} W^l h_j^{l-1} + b^l) \tag{4}$$

where W^l is a linear transformation, b^l is a bias, and σ denotes a nonlinear activation function, e.g., ReLU. $A \in \mathbb{R}^{n \times n}$ is obtained from the dependency tree, which is an adjacency matrix expressing connectivity between nodes.

However, directly stacking GCN and LSTM may cause a performance drop (Xu et al., 2021). We need to find a new solution to incorporate both types of features interaction between dependency trees and contextual information. Moreover, previous works for NER ignore dependency labels in the GCN modeling process.

3.3.2 AELGCN

Edge-Aware Node Joint Update Module Previous work (Cui et al., 2020) proposed an edge-aware node update (EANU) module that exploited the meaning of dependency labels in different contexts, and they did not consider that the dependency labels themselves may be incorrect. For EANU, this approach would cause the problem of remote propagation of incorrect dependency label information in node representations, finally, it will convey wrong information for NER and thus deteriorate the performance.

For this reason, we designed an edge-aware node joint update (EANJU) module. The EANJU module is able to mitigate the above problem. Theoretically, the EANJU module combine the structured information with dependency label information, via pool operation. If this dependency label information is incorrect, the structured information will be polluted. In order to mitigate this problem, we add its original structured information after pool operation to reduce the polluted influence.

Firstly, for a given dependency tree, we transform dependency tree into its corresponding adjacency matrix $A \in \mathbb{R}^{n \times n}$ and dependency label embedding adjacency matrix $E \in \mathbb{R}^{n \times n \times p}$ where $A_{ij} = 1$ indicates that node i and node j are connected, which means that node i and node j have dependency relation, $E_{i,j,:} \in \mathbb{R}^p$ denotes the p-dimensional dependency label representation between the node i and node j. With words in sentences interpreted as nodes in the graph, the EANJU module updates the representation for each node. Mathematically, this operation can be defined as follows:

$$\begin{aligned} H^l &= EANJU(E^{l-1}, A, H^{l-1}) \\ &= Pool(H_1^l, ..., H_p^l) + \sigma(AH^{l-1}W_1) \end{aligned} \tag{5}$$

Specifically, the aggregation is conducted channel

by channel in the adjacency tensor as follows:

$$H_i^l = E_{:,:,i}^{l-1} H^{l-1} W_2 \tag{6}$$

where $E \in \mathbb{R}^{n \times n \times p}$ is the dependency label embedding adjacency matrix from initialization or last AELGCN layer, $E_{:,:,i}^{l-1}$ denotes the i-th channel slice of E^{l-1}, H^0 is output of BiLSTM, $W_1 \in \mathbb{R}^{d \times d}$, $W_2 \in \mathbb{R}^{d \times d}$ are a learnable filter, d is the dimension of BiLSTM output representation and $A \in \mathbb{R}^{n \times n}$ is the adjacency matrix from initialization and σ is the ReLU activation function. A mean-pooling operation is applied to compress features since it covers information from all channels.

Node-Aware Edge Update Module Following the work by (Cui et al., 2020), it mentions that the same dependency label in different contexts may convey different signals, so specifying a context-independent representation for each dependency label is not sufficient to express the complex relationships between words. Therefore, (Cui et al., 2020) proposed a novel node-aware edge update (NAEU) module to dynamically calculate and update dependency label representations according to the node context. Formally, the NAEU operation is defined as:

$$\begin{aligned} E_{:,:,i}^l &= NAEU(E_{:,:,i}^{l-1}, h_i^l, h_j^l) \\ &= W_u[E_{:,:,i}^{l-1} \oplus h_i^l \oplus h_j^l] \end{aligned} \tag{7}$$

where $\oplus$ means the concatenation operator, h_i^l and h_i^l denote the representations of node i and j in the l_{th} layer after EANJU operation, $E_{:,:,i}^{l-1} \in \mathbb{R}^p$ is the relation representation between node i and j, $W_u \in \mathbb{R}^{2 \times d+p}$ is a learnable parameters. This updated dependency label embedding adjacency matrix is fed to the next AELGCN layer to perform another round of joint node updates, and such mutual update process can be stacked over L layers.

Attention Guided GCN In order to obtain syntactic information from different representation subspaces and learn how to attend to the relevant structured information selectively, we apply attention-guided GCN (AGGCN) (Guo et al., 2019) into our model. Unlike Vanilla GCN (Kipf and Welling, 2017), AGGCN will construct an attention-guided adjacency matrix $\tilde{A}$ generated by multi-head self-attention in AGGCN to update the node information again. The formula for generating $\tilde{A}$ is given as follows:

$$\tilde{A}^t = softmax(\frac{Q^t W_Q^t \times (K^t W_K^t)^T}{\sqrt{d_{head}}}) \tag{8}$$

where Q^t and K^t are both equal to EANJU output H^l or at layer $l-1$ of the AGGCN output h^{l-1}, W_Q^t and W_K^t are used to project the input $Q^t, K^t \in \mathbb{R}^{n \times d_{head}} (d_{head} = \frac{d_h}{N_{head}})$ of the t-th head into a query and a key $\tilde{A} \in \mathbb{R}^{n \times n}$ is the updated adjacency matrix for the t-th head.

For each head, AGGCN uses $\tilde{A}$ and a densely connected layer to deepen the layers of the whole AGGCN, to better capture the rich local information and k-hop information. The output of the densely connected layer is $\tilde{H}^t \in \mathbb{R}^{n \times d_h}$, then a linear combination layer is used to merge the output of each head, $\tilde{H} = [\tilde{H}^1, \tilde{H}^2, ..., \tilde{H}^{N_{head}}]W$, where $W \in \mathbb{R}^{(N_{head} \times d_h) \times d_h}$ is a learnable parameters, $\tilde{H} \in \mathbb{R}^{n \times d_h}$ is the final output of AGGCN.

After that, $\tilde{H}^t$ will be fed into the next layer of AELGCN to perform the same operation again and get the final output.

3.4 CRF Layer

We use a conditional random field (CRF) (Lafferty et al., 2001a) classifier at the top of our model to perform the sequential inference. The CRF takes vectors in the tagging space as input and produces the best sequence of labels using the Viterbi algorithm. Consider the observation sequence of vectors $x = [x_1, x_2, ..., x_n]$ and its corresponding target labels $y = [y_1, y_2, ..., y_n]$, CRF computes the conditional probability of the target sequence y given the inputs x by globally normalizing the target score:

$$P(y|x) = \frac{exp(score(x, y))}{\sum_{y'} exp(score(x, y'))} \tag{9}$$

The score function is defined as:

$$score(x, y) = \sum T_{y_i, y_{i+1}} + \sum E_{y_i} \tag{10}$$

where $T_{y_i, y_{i+1}}$ denotes the transition score from label y_i to y_{i+1}, E_{y_i} denotes the score of label y_i at the i_{th} position and the scores are computed using the hidden state. During training, we minimize the negative log-likelihood to obtain the model parameters.

4 Experiments

4.1 Datasets

Our proposed method is evaluated on four benchmark NER datasets: SemEval 2010 Task 1 (Recasens et al., 2010) Catalan and Spanish datasets,

Dataset		# Sent.	# Entity	# Entity Length					
				1	2	3	4	5	≥ 6
Catalan	Train	8,709	15,278	8,819	3,897	1,742	264	119	437
	Dev	1,445	2,431	1,370	676	269	40	18	58
	Test	1,698	2,910	1,601	811	338	57	27	76
Spanish	Train	9,022	17,297	10,307	3,609	2,302	301	175	603
	Dev	1,419	2,615	1,523	559	348	54	31	100
	Test	1,705	3,046	1,755	702	369	59	34	127
English	Train	59,924	81,828	46,525	17,391	9,714	4,892	1,938	1,368
	Dev	8,528	11,066	6,325	2,395	1,256	643	275	172
	Test	8,262	11,057	6,129	2,598	1,359	706	278	187
Chinese	Train	36,487	62,543	47,285	9,668	3,626	1,139	467	358
	Dev	6,083	9,104	6,969	1,397	473	169	55	41
	Test	4,472	7,494	5,479	1299	473	146	55	42

Table 1: Dataset statisticss.

and OntoNotes 5.0 (Weischedel et al., 2013) English and Chinese datasets. We chose these datasets because they contain both constituency tree and named entity annotations. For SemEval 2010 Task1 datasets, there are 4 entity types. For OntoNotes 5.0 datasets, there are 18 entity types in total. Following the work by (Xu et al., 2021), we transform the parse trees into the Stanford dependency trees (De Marneffe and Manning, 2008) by using Stanford CoreNLP (Manning et al., 2014). Moreover, we present the number of different lengths of entities to show that these datasets have a modest amount of long entities. Detailed statistics of each dataset can be found in Table 1.

4.2 Experimental Setup

For Catalan and Spanish, we use Subs2Vec (Paridon and Thompson, 2020) 100-dimensional embeddings to initialize the word embeddings. For OntoNotes 5.0 Chinese, we use SGNS Word2vec (Qiu et al., 2018) 300-dimensional embeddings to initialize the word embeddings. For OntoNotes 5.0 English, we adopt the publicly available GloVE (Pennington et al., 2014) 100-dimensional embeddings to initialize the word embeddings. For experiments with the contextualized representation, we adopt the pre-trained language model BERT (Devlin et al., 2019) for the four datasets. We use the cased version of the BERT large model for the OntoNotes 5.0 English data experiments. We use the cased version of the BERT base model for the experiments on the other three datasets. For the character embeddings, we randomly initialize the character embeddings, set the dimension as 30, and set the hidden size of character-level BiLSTM as 50. For the dependency label embeddings, we randomly initialize dependency label embeddings with 50-dimension vectors and dependency label embedding adjacency matrix embeddings as 50. The hidden size of AELGCN and BiLSTM is set as 200, and the number of AELGCN layers L as 2. For AGGCN, we set the number of heads for the attention guided layer as 4, the first block number as 2, and the number of sublayers L in each densely connected layer as 4. Our models are optimized by mini-batch stochastic gradient descent (SGD) with a learning rate of 0.1 and batch size of 20. We use $L2$ regularization with a parameter of 1e-8 to avoid overfitting. Dropout is applied to word embeddings and hidden states with a rate of 0.5. We ran experiments using Pytorch 1.9.0 on Nvidia Tesla K40m GPU with Intel Xeon E5-2620 CPU.

4.3 Baselines

We compare our models with several competitive dependency-based models.

- BiLSTM-GCN-CRF (Jie and Lu, 2019), which simply stacks GCN on top of BiLSTM to incorporate the dependency trees.
- Dependency guided LSTM-CRF (DGLSTM-CRF) (Jie and Lu, 2019), which takes the concatenation of head word representation and word embeddings as input into BiLSTM.
- GCN-BiLSTM-CRF (Xu et al., 2021), which takes the concatenation of the graph-encoded representation from GCN and word embedding as input into BiLSTM.
- Syn-LSTM-CRF (Xu et al., 2021), a recurrent neural network architecture considers an additional graph-encoded representation to update the memory and hidden states.

Besides, we compare our model with previous works that have results on these datasets.

4.4 Results

SemEval 2010 Task 1 Table 2 shows the comparison of our model with the baseline models on the SemEval 2010 Task 1 Catalan and Spanish datasets. Our BiLSTM-AELGCN-CRF model outperforms all models with F_1 86.75 and 88.13. Our model outperforms the BiLSTM-CRF model by 17.24 and 14.26 percentage points in F_1, outperforms the BiLSTM-GCN-CRF model by 11.53 and 6.20 F_1 points, and outperforms the GCN-BiLSTM-CRF model 9.32 and 6.28 F_1. Compared to the DGLSTM-CRF, our proposed method improves 5.11 and 4.66 F_1 points on the Catalan and Spanish datasets. In addition, compared to Syn-LSTM-CRF, we improved 3.99 and 3.04 F_1 points, respectively.

Models	Catalan			Spanish		
	P.	*R.*	F_1	*P.*	*R.*	F_1
BiLSTM-CRF*	76.83	63.47	69.51	78.33	69.89	73.87
BiLSTM-GCN-CRF*	81.25	75.22	78.12	84.10	79.88	81.93
GCN-BiLSTM-CRF†	80.95	74.19	77.43	84.36	79.48	81.85
DGLSTM-CRF (2019)	83.35	80.00	81.64	84.05	82.90	83.47
Syn-LSTM-CRF (2021)	83.90	81.65	82.76	86.22	84.24	85.09
BiLSTM-AELGCN-CRF (Ours)	**87.60**	**85.91**	**86.75**	**88.75**	**87.52**	**88.13**
Improvement Δ	+3.70	+4.26	+3.99	+2.53	+3.29	+3.04
+ Contextualized Word Representation						
BERT-CRF†	76.34	76.05	76.19	79.30	77.22	78.24
Wolf et al. (2020)†	83.42	85.7	84.23	81.36	85.58	83.42
BiLSTM-CRF $_{+\text{ ELMO}}$*	77.85	76.22	77.03	81.72	79.09	80.38
BiLSTM-CRF $_{+\text{ BERT}}$†	81.21	79.90	80.55	83.28	80.11	81.66
BiLSTM-GCN-CRF$_{+\text{ ELMO}}$*	83.68	83.16	83.42	85.31	85.19	85.25
GCN-BiLSTM-CRF$_{+\text{ BERT}}$†	87.60	86.39	86.99	88.07	87.46	87.76
DGLSTM-CRF (2019) $_{+\text{ ELMO}}$	84.71	83.75	84.22	87.79	87.33	87.56
DGLSTM-CRF$_{+\text{ BERT}}$†	85.92	84.50	85.20	85.67	85.00	85.33
Syn-LSTM-CRF (2021) $_{+\text{ BERT}}$	89.07	89.04	89.05	89.66	**90.54**	90.10
BiLSTM-AELGCN-CRF$_{+\text{ BERT}}$ (Ours)	**90.11**	**90.21**	**90.16**	**91.86**	90.41	**91.13**
Improvement Δ	+1.04	+1.17	+1.11	+2.20	-0.11	+1.03

Table 2: Experimental results [%] on SemEval 2010 Task 1 Catalan and Spanish test set. The models with * symbol are retrieved from (Jie and Lu, 2019) and † symbol are retrieved from (Xu et al., 2021)

The results show that our proposed model can effectively capture structured information compared to traditional GCN.

We further compare the performance of all models with contextualized word representation, BiLSTM-AELGCN-CRF$_{+\text{ BERT}}$ achieves higher performance improvements than any other method. Our model outperformed the strong baseline model Syn-LSTM-CRF$_{+\text{ BERT}}$ in F_1 by 1.11 and 1.03 in Catalan and Spanish, respectively.

OntoNotes 5.0 English Table 3 shows the performance comparison between our work and previous work on the OntoNotes English 5.0 dataset. Our BiLSTM-AELGCN-CRF model outperforms all existing methods with 89.25 in terms of F_1 score. Our model outperforms the BiLSTM-CRF model by 2.18 in F_1 and the BiLSTM-GCN-CRF model by 1.07. Compared to the DGLSTM-CRF and Syn-LSTM-CRF, our proposed method improves 0.73 and 0.21 F_1 points, respectively. Although our proposed method that precision drops compared to Syn-LSTM-CRF, the performance improvement on recall is more significant. This shows that our proposed method is able to extract more entities. Moreover, with the contextualized word representation BERT, our method achieves an F_1 score of 91.16. Our method outperforms Syn-LSTM-CRF model by 0.31 F_1.

OntoNotes 5.0 Chinese Our experimental results on OntoNotes 5.0 Chinese test set are shown in Table 4. Our model still consistently outperforms the baseline models. Our model outperforms

Models	*P.*	*R.*	F_1
Chiu and Nichols (2016)	86.04	86.53	86.28
Li et al. (2017)	88.00	86.50	87.21
Strubell et al. (2017)	-	-	86.84
Ghaddar and Langlais (2018)	-	-	87.95
BiLSTM-CRF*	87.21	86.93	87.07
BiLSTM-GCN-CRF*	88.30	88.06	88.18
GCN-BiLSTM-CRF†	88.56	88.76	88.66
DGLSTM-CRF (2019)	88.53	88.50	88.52
Luo et al. (2020)	-	-	87.98
Syn-LSTM-CRF (2021)	**88.96**	89.13	89.04
BiLSTM-AELGCN-CRF (Ours)	88.72	**89.79**	**89.25**
Improvement Δ	-0.24	+0.66	+0.21
+ Contextualized Word Representation			
Akbik et al. (2018)	-	-	89.30
BERT-CRF†	88.42	88.33	88.37
Wolf et al. (2020)†	88.39	90.29	89.33
BiLSTM-CRF$_{+\text{ ELMO}}$*	89.14	88.59	88.87
BiLSTM-CRF$_{+\text{ BERT}}$†	89.32	90.02	89.67
BiLSTM-GCN-CRF$_{+\text{ ELMO}}$*	89.40	89.71	89.55
GCN-BiLSTM-CRF$_{+\text{ BERT}}$†	89.34	91.26	90.29
DGLSTM-CRF(2019)$_{+\text{ ELMO}}$	89.59	90.17	89.88
DGLSTM-CRF$_{+\text{ BERT}}$†	89.63	89.87	89.75
Luo et al. (2020)$_{+\text{ BERT}}$*	-	-	90.30
Syn-LSTM-CRF (2021)$_{+\text{ BERT}}$	90.14	91.58	90.85
BiLSTM-AELGCN-CRF$_{+\text{ BERT}}$ (Ours)	**90.55**	**91.78**	**91.16**
Improvement Δ	+0.41	+0.20	+0.31

Table 3: Experimental results [%] on OntoNotes 5.0 English test set. The models with * symbol are retrieved from (Jie and Lu, 2019) and † symbol are retrieved from (Xu et al., 2021).

the BiLSTM-CRF model by 2.97 in F_1, and the BiLSTM-GCN-CRF model by 3.32 in F_1. Note that the BiLSTM-GCN-CRF model is 0.35 points worse than BiLSTM-CRF. This confirms that simply stacking GCN on top of the BiLSTM does not perform well, which may cause a performance drop. Compared to the DGLSTM-CRF and Syn-LSTM-CRF, our proposed method improves 2.04 and 0.93 F_1 points, respectively. With the contextualized word representation, we achieve a higher F_1 score of 80.89. However, it is worth noting that the dependency-based models with the contextualized word representation have different degrees of decline in precision compared to BERT-CRF. The reason could be that some of the entities do not form subtrees under the dependency trees. In such a situation, the model with the contextualized word representation may not correctly identify the boundary of the entities, which results in lower precision.

5 Analysis

Ablation Study To demonstrate the effectiveness of each component, we conduct an ablation study

Models	*P.*	*R.*	F_1
Pradhan et al. (2013)	78.20	66.45	71.85
Lattice LSTM (2018)	76.34	77.01	76.67
BiLSTM-CRF*	78.45	74.59	76.47
BiLSTM-GCN-CRF*	76.35	75.89	76.12
GCN-BiLSTM-CRF†	78.30	75.07	76.65
DGLSTM-CRF (2019)	77.40	77.41	77.40
Syn-LSTM-CRF (2021)	77.95	79.07	78.51
BiLSTM-AELGCN-CRF (Ours)	**79.11**	**79.78**	**79.44**
Improvement Δ	+1.16	+0.71	+0.93
+ Contextualized Word Representation			
BERT-CRF†	**79.83**	79.68	79.75
Wolf et al. (2020)†	77.35	81.74	79.49
BiLSTM-CRF$_{+\text{ ELMO}}$*	79.20	79.21	79.20
BiLSTM-CRF$_{+\text{ BERT}}$†	78.45	81.24	79.82
BiLSTM-GCN-CRF$_{+\text{ ELMO}}$*	78.71	79.29	79.00
GCN-BiLSTM-CRF$_{+\text{ BERT}}$†	79.03	80.98	80.00
DGLSTM-CRF (2019)$_{+\text{ ELMO}}$	78.86	81.00	79.92
DGLSTM-CRF$_{+\text{ BERT}}$†	77.79	81.65	79.67
Syn-LSTM-CRF (2021)$_{+\text{ BERT}}$	78.66	81.80	80.20
BiLSTM-AELGCN-CRF$_{+\text{ BERT}}$ (Ours)	79.23	**82.63**	**80.89**
Improvement Δ	-0.60	+0.83	+0.69

Table 4: Experimental results [%] on OntoNotes 5.0 Chinese test set. The models with * symbol are retrieved from (Jie and Lu, 2019) and † symbol are retrieved from (Xu et al., 2021).

Model	Catalan	Spanish	English	Chinese
	F$_1$-score(%)			
BiLSTM-AELGCN-CRF	**86.75**	**88.13**	**89.25**	**79.44**
– AGGCN	86.36	87.92	89.05	79.18
– vanilla GCN in EANJU	85.50	87.43	88.99	79.24
– EANJU	81.87	83.54	88.47	79.03
– NAEU	86.50	87.93	89.11	79.22

Table 5: Ablation study of the BiLSTM-ALEGCN-CRF model on four datasets. – means removing.

Dataset	Model	Entity Length					
		1	**2**	**3**	**4**	**5**	**≥ 6**
Catalan	BiLSTM-CRF$_{+\text{ BERT}}$	82.4	84.4	77.8	53.3	31.8	36.2
	DGLSTM-CRF$_{+\text{ ELMO}}$	85.4	85.1	84.1	78.9	60.9	59.3
	Syn-LSTM-CRF$_{+\text{ BERT}}$	90.5	91.1	87.2	77.8	63.8	60.6
	BiLSTM-AELGCN-CRF$_{+\text{ BERT}}$	**91.1**	**91.6**	**89.0**	**84.0**	**73.0**	**69.5**
Spanish	BiLSTM-CRF$_{+\text{ BERT}}$	85.1	84.2	81.5	33.7	43.1	27.2
	DGLSTM-CRF$_{+\text{ ELMO}}$	89.3	87.4	90.8	**74.1**	67.7	**64.4**
	Syn-LSTM-CRF$_{+\text{ BERT}}$	92.7	90.9	91.1	73.0	75.4	58.5
	BiLSTM-AELGCN-CRF$_{+\text{ BERT}}$	**93.3**	**91.5**	**92.1**	73.0	**77.4**	62.2
Chinese	BiLSTM-CRF$_{+\text{ BERT}}$	82.5	74.6	71.4	65.0	**69.8**	**52.5**
	DGLSTM-CRF$_{+\text{ ELMO}}$	82.2	75.5	71.8	64.1	58.5	41.1
	Syn-LSTM-CRF$_{+\text{ BERT}}$	82.5	75.6	73.1	66.4	66.1	42.5
	BiLSTM-AELGCN-CRF$_{+\text{ BERT}}$	**83.2**	**76.3**	**74.3**	**67.6**	69.2	44.1
English	BiLSTM-CRF$_{+\text{ BERT}}$	92.9	88.3	83.1	85.5	80.5	77.9
	DGLSTM-CRF$_{+\text{ ELMO}}$	91.8	90.1	85.4	87.0	80.8	78.7
	Syn-LSTM-CRF$_{+\text{ BERT}}$	92.9	90.8	87.7	87.4	80.6	79.8
	BiLSTM-AELGCN-CRF$_{+\text{ BERT}}$	**92.9**	**91.1**	**87.9**	**87.7**	**84.3**	**82.2**

Table 6: F$_1$-score [%] based on entity length on Catalan, Spanish, English and Chinese datasets. Note that DGLSTM-CRF$_{+\text{ ELMO}}$ have better performance compared to DGLSTM-CRF$_{+\text{ BERT}}$ based on the results.

on each of the four benchmark datasets as Table 5 shows 1) - AGGCN: we remove the AGGCN then we observe that the performance reduces by 0.39, 0.21, 0.26, and 0.20 of the F_1 scores on the results, respectively, which demonstrates that the AGGCN captures useful syntactic information. 2) - vanilla GCN: we remove the vanilla GCN in EANJU, which means that EANJU degenerates into the EANU (Cui and Chen, 2022). As a result, the F_1 scores drop by 1.25, 0.70, 0.20, and 0.26, respectively, demonstrating that EANJU captures more useful syntactic information than EANU. 3) - EANJU: the results drop by 4.88, 4.59, 0.41, and 0.78 F_1 scores, respectively, which demonstrates that syntactic structure information plays an important role in the model. 4) - NAEU: the results drop by 0.25, 0.20, 0.22, and 0.14 of F_1 scores, respectively, which verifies that the context-dependent relation representations also provide some useful information for NER than the context-independent ones.

Effect of Entity Length Table 6 shows the performance of NER comparison with different entity lengths on all datasets. We compare the BiLSTM-CRF$_{+\text{ BERT}}$, DGLSTM-CRF$_{+\text{ ELMO}}$, Syn-LSTM-CRF$_{+\text{ BERT}}$ and BiLSTM-AELGCN-CRF$_{+\text{ BERT}}$ models with respect to entity length $\in \{1, 2, 3, 4, 5, \geq 6\}$ on the four languages. With the structured information, DGLSTM-CRF$_{+\text{ ELMO}}$, Syn-LSTM-CRF$_{+\text{ BERT}}$ and BiLSTM-AELGCN-CRF$_{+\text{ BERT}}$ models achieve better performance compared to BiLSTM-CRF$_{+\text{ BERT}}$. When the length of entity is ≤ 3, BiLSTM-AELGCN-CRF$_{+\text{ BERT}}$ achieves better results compared to DGLSTM-CRF$_{+\text{ ELMO}}$ and Syn-LSTM-CRF$_{+\text{ BERT}}$. Our model consistently outperforms Syn-LSTM-CRF$_{+\text{ BERT}}$. These results confirm that our proposed method can effectively incorporate structured information . When the length of an entity is equal to or longer than 4, there may be internal dependencies (subtree) between the words in the entity, which can provide valuable information to improve the performance of dependency-based models. Thus, all dependency-based models perform much better than BiLSTM-CRF$_{+\text{ BERT}}$ on all datasets except OntoNotes Chinese dataset. This exceptional case may due to the fact that the ratio of entities that form subtrees in this dataset is relatively smaller compared to other datasets, 92.9% versus nearly 100% (Jie and Lu, 2019). But our proposed method also performs better than Syn-LSTM-CRF$_{+\text{ BERT}}$ and DGLSTM-CRF$_{+\text{ ELMO}}$ on

OntoNotes Chinese dataset. In general, our proposed method slightly improves performance on short entities compared to other models. Further, our proposed method is more effective for long entities than other dependency-based models in most cases, especially for the Catalan dataset.

Impact of AELGCN layers As AELGCN can be stacked over L layers, we investigate the effect of the layer number L on the final performance. We conduct another experiment on the BiLSTM-AELGCN-CRF model with the number of AELGCN layers $\in \{1, 2, 3\}$ on test datasets. The last AVG bar is obtained by averaging the results of the four test datasets. As shown in Figure 3, the performance increases as the number of AELGCN layers increases from 1 to 2 and decreases when the number of layers increases from 2 to 3. The latter phenomenon could be caused by an over-smooth problem of deep GCNs, which also exists in (Xu et al., 2021) for named entity recognition and (Kipf and Welling, 2017) for document classification and node classification. For this observation, it is considered that when $L = 1$, AELGCN can only utilize first-order syntactic relations on the dependency tree, which is insufficient to bring important contextual words with multiple hops on the dependency tree into the entity recognition. Therefore, we evaluate our proposed BiLSTM-AELGCN-CRF model with 2-layer AELGCN.

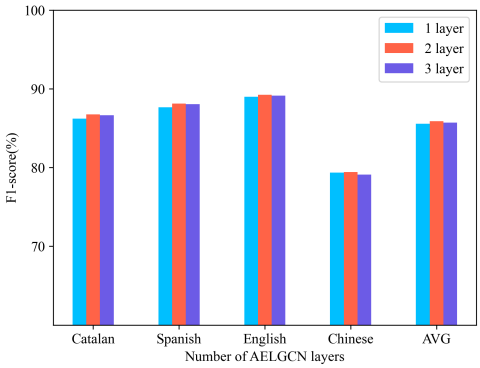

Figure 3: F_1-score [%] variation with AELGCN layers on test datasets.

6 Conclusion and Future Works

In this paper, we propose a novel model named BiLSTM-AELGCN-CRF for the NER task. Specifically, we introduce the dependency label information and multi-head self-attention mechanism into the graph modeling process. Our analysis shows that our method can better capture structured information which is beneficial for the model to recognize entities.

In the future, we would like to apply BiLSTM-AELGCN-CRF to other information extraction tasks, such as relation extraction or joint entity and relation extraction. Moreover, we will continue to explore how to use syntactic information better for NER tasks.

Limitations

The limitation of our model is that the performance of our model is highly dependent on the quality of the dependency trees. In most cases, the quality of the automatically generated dependency trees is good enough for our model. However, in some cases, the dependency trees generated by automatic tools are lack of sufficient and high quality dependency information. Under such cases, the performance of our method will be greatly decreased by the insufficient or poor-quality dependency information, becomes even worse than that of dependency-tree-free methods. This problem can be seen from the result of ontonoes Chinese dataset in table 6. After investigation, it is found that the percentage of entities that have subtrees is only 92.9% for OntoNotes Chinese dataset, as compared to 98.5%, 100%, 100% for OntoNotes English, SemEval Catalan and Spanish, respectively (Jie and Lu, 2019).

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments and suggestions on this work. This work is supported by National Key Research and Development Program of China under grant No.2019YFB2102100; The National Natural Science Foundation of China under Grant (No.62072146, No.61972358); The Key Research and Development Program of Zhejiang Province (2021C03145, 2019C03134).

References

Gustavo Aguilar and Thamar Solorio. 2019. Dependency-aware named entity recognition with relative and global attentions. *arXiv preprint arXiv:1909.05166*.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence label-

ing. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yoshua Bengio. 2009. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127.

Alberto Cetoli, Stefano Bragaglia, Andrew O'Harney, and Marc Sloan. 2017. Graph convolutional networks for named entity recognition. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, pages 37–45, Prague, Czech Republic.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Shiyao Cui, Bowen Yu, Tingwen Liu, Zhenyu Zhang, Xuebin Wang, and Jinqiao Shi. 2020. Edge-enhanced graph convolution networks for event detection with syntactic relation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2329–2339, Online. Association for Computational Linguistics.

Wanyun Cui and Xingran Chen. 2022. Enhancing natural language representation with large-scale out-of-domain commonsense. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1746–1756, Dublin, Ireland. Association for Computational Linguistics.

Marie-Catherine De Marneffe and Christopher D Manning. 2008. Stanford typed dependencies manual. Technical report, Technical report, Stanford University.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Abbas Ghaddar and Phillippe Langlais. 2018. Robust lexical features for improved neural network named-entity recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1896–1907, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251, Florence, Italy. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Zhanming Jie and Wei Lu. 2019. Dependency-guided LSTM-CRF for named entity recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3862–3872, Hong Kong, China. Association for Computational Linguistics.

Zhanming Jie, Aldrian Obaja Muis, and Wei Lu. 2017. Efficient dependency-guided named entity recognition. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3457–3465. AAAI Press.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001a. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001b. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Peng-Hsuan Li, Ruo-Ping Dong, Yu-Siang Wang, Ju-Chieh Chou, and Wei-Yun Ma. 2017. Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2664–2669, Copenhagen, Denmark. Association for Computational Linguistics.

Zhifei Li, Hai Liu, Zhaoli Zhang, Tingting Liu, and Neal N. Xiong. 2022. Learning knowledge graph embedding with heterogeneous relation attention networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(8):3961–3973.

Xiao Liu, Heyan Huang, and Yue Zhang. 2019. Open domain event extraction using neural latent variable models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2860–2871, Florence, Italy. Association for Computational Linguistics.

Ying Luo, Fengshun Xiao, and Hai Zhao. 2020. Hierarchical contextualized representation for named entity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8441–8448.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Jeroen Paridon and Bill Thompson. 2020. subs2vec: Word embeddings from subtitles in 55 languages. *Behavior Research Methods*, 53.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Yuanyuan Qiu, Hongzheng Li, Shen Li, Yingdi Jiang, Renfen Hu, and Lijiao Yang. 2018. Revisiting correlations between intrinsic and extrinsic evaluations of word embeddings. In *Chinese computational linguistics and natural language processing based on naturally annotated big data*, pages 209–221. Springer.

Marta Recasens, Lluís Màrquez, Emili Sapena, M. Antònia Martí, Mariona Taulé, Véronique Hoste, Massimo Poesio, and Yannick Versley. 2010. SemEval-2010 task 1: Coreference resolution in multiple languages. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 1–8, Uppsala, Sweden. Association for Computational Linguistics.

Stefan Schweter and Alan Akbik. 2020. Flert: Document-level features for named entity recognition. *CoRR*, abs/2011.06993.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680, Copenhagen, Denmark. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing.

In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Lu Xu, Zhanming Jie, Wei Lu, and Lidong Bing. 2021. Better feature integration for named entity recognition. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3457–3469, Online. Association for Computational Linguistics.

Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with Freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 956–966, Baltimore, Maryland. Association for Computational Linguistics.

Yue Zhang and Jie Yang. 2018. Chinese NER using lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1564, Melbourne, Australia. Association for Computational Linguistics.