

Dependency-Guided LSTM-CRF for Named Entity Recognition

Zhanming Jie and Wei Lu

StatNLP Research Group

Singapore University of Technology and Design

zhanming_jie@mymail.sutd.edu.sg, luwei@sutd.edu.sg

Abstract

Dependency tree structures capture long-distance and syntactic relationships between words in a sentence. The syntactic relations (e.g., *nominal subject*, *object*) can potentially infer the existence of certain named entities. In addition, the performance of a named entity recognizer could benefit from the long-distance dependencies between the words in dependency trees. In this work, we propose a simple yet effective *dependency-guided* LSTM-CRF model to encode the complete dependency trees and capture the above properties for the task of named entity recognition (NER). The data statistics show strong correlations between the entity types and dependency relations. We conduct extensive experiments on several standard datasets and demonstrate the effectiveness of the proposed model in improving NER and achieving state-of-the-art performance. Our analysis reveals that the significant improvements mainly result from the dependency relations and long-distance interactions provided by dependency trees.

1 Introduction

Named entity recognition (NER) is one of the most important and fundamental tasks in natural language processing (NLP). Named entities capture useful semantic information which was shown helpful for downstream NLP tasks such as coreference resolution (Lee et al., 2017), relation extraction (Miwa and Bansal, 2016) and semantic parsing (Dong and Lapata, 2018). On the other hand, dependency trees also capture useful semantic information within natural language sentences. Currently, research efforts have derived useful discrete features from dependency structures (Sasano and Kurohashi, 2008; Cucchiarelli and Velardi, 2001; Ling and Weld, 2012) or structural constraints (Jie et al., 2017) to help the NER task. However, how

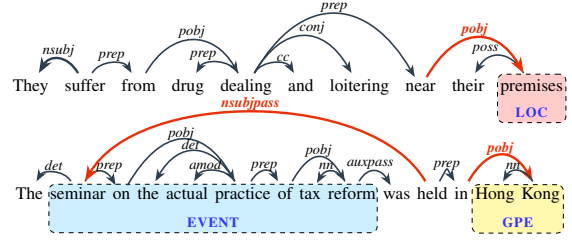


Figure 1: Example sentences annotated with named entities and dependencies in the OntoNotes 5.0 dataset.

to make good use of the rich relational information as well as complex long-distance interactions among words as conveyed by the complete dependency structures for improved NER remains a research question to be answered.

The first example in Figure 1 illustrates the relationship between a dependency structure and a named entity. Specifically, the word “*premises*”, which is a named entity of type LOC (location), is characterized by the incoming arc with label “*pobj*” (prepositional object). This arc reveals a certain level of the semantic role that the word “*premises*” plays in the sentence. Similarly, the two words “*Hong Kong*” in the second example that form an entity of type GPE are also characterized by a similar dependency arc towards them.

The long-distance dependencies capturing non-local structural information can also be very helpful for the NER task (Finkel et al., 2005). In the second example of Figure 1, the long-distance dependency from “*held*” to “*seminar*” indicates a direct relation “*nsubjpass*” (passive subject) between them, which can be used to characterize the existence of an entity. However, existing NER models based on linear-chain structures would have difficulties in capturing such long-distance relations (i.e., non-local structures).

One interesting property, as highlighted in the work of Jie et al. (2017), is that most of the en-

entities form subtrees under their corresponding dependency trees. In the example of the EVENT entity in Figure 1, the entity itself forms a subtree and the words inside have rich complex dependencies among themselves. Exploiting such dependency edges within the subtrees allows a model to capture non-trivial semantic-level interactions between words within long entities. For example, “practice” is the prepositional object (*pobj*) of “on” which is a preposition (*prep*) of “seminar” in the EVENT entity. Modeling these grandchild dependencies (GD) (Koo and Collins, 2010) requires the model to capture some higher-order long-distance interactions among different words in a sentence.

Inspired by the above characteristics of dependency structures, in this work, we propose a simple yet effective dependency-guided model for NER. Our neural network based model is able to capture both contextual information and rich long-distance interactions between words for the NER task. Through extensive experiments on several datasets on different languages, we demonstrate the effectiveness of our model, which achieves the state-of-the-art performance. To the best of our knowledge, this is the first work that leverages the complete dependency graphs for NER. We make our code publicly available at <http://www.statnlp.org/research/information-extraction>.

2 Related Work

NER has been a long-standing task in the field of NLP. While many recent works (Peters et al., 2018a; Akbik et al., 2018; Devlin et al., 2019) focus on finding good contextualized word representations for improving NER, our work is mostly related to the literature that focuses on employing dependency trees for improving NER.

Sasano and Kurohashi (2008) exploited the syntactic dependency features for Japanese NER and achieved improved performance with a support vector machine (SVM) (Cortes and Vapnik, 1995) classifier. Similarly, Ling and Weld (2012) included the head word in a dependency edge as features for fine-grained entity recognition. Their approach is a pipeline where they extract the entity mentions with linear-chain conditional random fields (CRF) (Lafferty et al., 2001) and used a classifier to predict the entity type. Liu et al. (2010) proposed to link the words that are associ-

ated with selected typed dependencies (e.g., “nn”, “prep”) using a skip-chain CRF (Sutton and McCallum, 2004) model. They showed that some specific relations between the words can be exploited for improved NER. Cucchiarelli and Velardi (2001) applied a dependency parser to obtain the syntactic relations for the purpose of unsupervised NER. The resulting relation information serves as the features for potential existence of named entities. Jie et al. (2017) proposed an efficient dependency-guided model based on the semi-Markov CRF (Sarawagi and Cohen, 2004) for NER. The purpose is to reduce time complexity while maintaining the non-Markovian features. They observed certain relationships between the dependency edges and the named entities. Such relationships are able to define a reduced search space for their model. While these previous approaches do not make full use of the dependency tree structures, we focus on exploring neural architectures to exploit the complete structural information conveyed by the dependency trees.

3 Model

Our dependency-guided model is based on the state-of-the-art BiLSTM-CRF model proposed by Lample et al. (2016). We first briefly present their model as background and next present our dependency-guided model.

3.1 Background: BiLSTM-CRF

In the task of named entity recognition, we aim to predict the label sequence $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$ given the input sentence $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ where n is the number of words. The labels in \mathbf{y} are defined by a label set with the standard IOBES¹ labeling scheme (Ramshaw and Marcus, 1999; Ratnoff and Roth, 2009). The CRF (Lafferty et al., 2001) layer defines the probability of the label sequence \mathbf{y} given \mathbf{x} :

$$P(\mathbf{y}|\mathbf{x}) = \frac{\exp(\text{score}(\mathbf{x}, \mathbf{y}))}{\sum_{\mathbf{y}'} \exp(\text{score}(\mathbf{x}, \mathbf{y}'))} \quad (1)$$

Following Lample et al. (2016), the score is defined as the sum of transitions and emissions from the bidirectional LSTM (BiLSTM):

$$\text{score}(\mathbf{x}, \mathbf{y}) = \sum_{i=0}^n A_{y_i, y_{i+1}} + \sum_{i=1}^n F_{\mathbf{x}, y_i} \quad (2)$$

¹“S-” indicates the entity with a single word and “E-” indicates the end of an entity.

where \mathbf{A} is a transition matrix in which $A_{y_i, y_{i+1}}$ is the transition parameter from the label y_i to the label y_{i+1} ². \mathbf{F}_x is an emission matrix where F_{x, y_i} represents the scores of the label y_i at the i -th position. Such scores are provided by the parameterized LSTM (Hochreiter and Schmidhuber, 1997) networks. During training, we minimize the negative log-likelihood to obtain the model parameters including both LSTM and transition parameters.

3.2 Dependency-Guided LSTM-CRF

Input Representations The word representation \mathbf{w} in the BiLSTM-CRF (Lample et al., 2016; Ma and Hovy, 2016; Reimers and Gurevych, 2017) model consists of the concatenation of the word embedding as well as the corresponding character-based representation. Inspired by the fact that each word (except the root) in a sentence has exactly one *head* (i.e., *parent*) word in the dependency structure, we can enhance the word representations with such dependency information. Similar to the work by Miwa and Bansal (2016), we concatenate the word representation together with the corresponding head word representation and dependency relation embedding as the input representation. Specifically, given a dependency edge (x_h, x_i, r) with x_h as parent, x_i as child and r as dependency relation, the representation at position i can be denoted as:

$$\mathbf{u}_i = [\mathbf{w}_i; \mathbf{w}_h; \mathbf{v}_r], \quad x_h = \text{parent}(x_i) \quad (3)$$

where \mathbf{w}_i and \mathbf{w}_h are the word representations of the word x_i and its parent x_h , respectively. We take the final hidden state of character-level BiLSTM as the character-based representation (Lample et al., 2016). \mathbf{v}_r is the embedding for the dependency relation r . These relation embeddings are randomly initialized and fine-tuned during training. The above representation allows us to capture the direct long-distance interactions at the input layer. For the word that is a root of the dependency tree, we treat its parent as itself³ and create a root relation embedding. Additionally, contextualized word representations (e.g., ELMo) can also be concatenated into \mathbf{u} .

Neural Architecture Given the dependency-encoded input representation \mathbf{u} , we apply the LSTM to capture the contextual information and

² y_0 and y_{n+1} are start and end labels.

³We also tried using a root word embedding but the performance is similar.

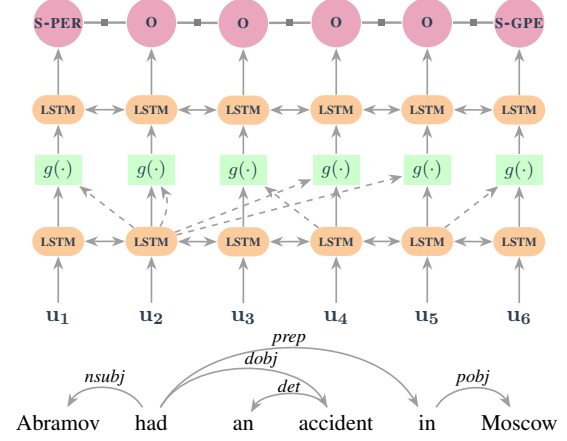


Figure 2: Dependency-guided LSTM-CRF with 2 LSTM Layers. Dashed connections mimic the dependency edges. “ $g(\cdot)$ ” represents the interaction function.

model the interactions between the words and their corresponding parents in the dependency trees. Figure 2 shows the proposed dependency-guided LSTM-CRF (**DGLSTM-CRF**) with 2 LSTM layers for the example sentence “Abramov had an accident in Moscow” and its dependency structure. The corresponding label sequence is $\{S\text{-PER}, O, O, O, O, S\text{-GPE}\}$. Followed by the first BiLSTM, the hidden states at each position will propagate to the next BiLSTM layer and its child along the dependency trees. For example, the hidden state of the word “had”, $\mathbf{h}_2^{(1)}$, will propagate to its child “Abramov” at the first position. For the word that is root, the hidden state at that specific position will propagate to itself. We use an *interaction function* $g(\mathbf{h}_i, \mathbf{h}_{p_i})$ to capture the interaction between the child and its parent in a dependency. Such an interaction function can be concatenation, addition or a multilayer perceptron (MLP). We further apply another BiLSTM layer on top of the interaction functions to produce the context representation for the final CRF layer.

The architecture shown in Figure 2 with a 2-layer BiLSTM can effectively encode the grandchild dependencies because the input representations encode the parent information and the interaction function further propagate the grandparent information. Such propagations allow the model to capture the indirect long-distance interactions from the grandchild dependencies between the words in the sentence as mentioned in Section 1. In general, we can stack more interaction functions and BiLSTMs to enable deeper reasoning over the dependency trees. Specifically, the hid-

Interaction Function	$g(\mathbf{h}_i, \mathbf{h}_{p_i})$
Self connection	\mathbf{h}_i
Concatenation	$\mathbf{h}_i \oplus \mathbf{h}_{p_i}$
Addition	$\mathbf{h}_i + \mathbf{h}_{p_i}$
MLP	$\text{ReLU}(\mathbf{W}_1 \mathbf{h}_i + \mathbf{W}_2 \mathbf{h}_{p_i})$

Table 1: List of interaction functions.

den states of the $(l + 1)$ -th layer $\mathbf{H}^{(l+1)}$ can be calculated from the hidden state of the previous layer $\mathbf{H}^{(l)}$:

$$\begin{aligned}\mathbf{H}^{(l+1)} &= \text{BiLSTM}\left(f(\mathbf{H}^{(l)})\right) \\ \mathbf{H}^{(l)} &= [\mathbf{h}_1^{(l)}, \mathbf{h}_2^{(l)}, \dots, \mathbf{h}_n^{(l)}] \\ f(\mathbf{H}^{(l)}) &= [g(\mathbf{h}_1^{(l)}, \mathbf{h}_{p_1}^{(l)}), \dots, g(\mathbf{h}_n^{(l)}, \mathbf{h}_{p_n}^{(l)})]\end{aligned}$$

where p_i indicates the parent index of the word x_i . $g(\mathbf{h}_i^{(l)}, \mathbf{h}_{p_i}^{(l)})$ represents the interaction functions between the hidden state at the i -th and p_i -th positions under the dependency edges (x_{p_i}, x_i) . The number of layers L can be chosen according to the performance on the development set.

Interaction Function The interaction function between the parent and child representations can be defined in various ways. Table 1 shows the list of interaction function considered in our experiments. The first one returns the hidden state itself, which is equivalent to stacking the LSTM layers. The concatenation and addition involve no parameter, which are straightforward ways to model the interactions. The last one applies an MLP to model the interaction between parent and child representations. With the rectified linear unit (ReLU) as activation function, the $g(\mathbf{h}_i, \mathbf{h}_{p_i})$ function is analogous to a graph convolutional network (GCN) (Kipf and Welling, 2017) formulation. In such a graph, each node has a self connection (i.e., \mathbf{h}_i) and a dependency connection with parent (i.e., \mathbf{h}_{p_i}). Similar to the work by Marcheggiani and Titov (2017), we adopt different parameters \mathbf{W}_1 and \mathbf{W}_2 for self and dependency connections.

4 Experiments

Datasets The main experiments are conducted on the large-scale 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. There are 18 types of entities defined in the OntoNotes dataset. We convert the constituency

Dataset	Train		Dev		Test		ST	GD
	# Sent.	# Entity	# Sent.	# Entity	# Sent.	# Entity	(%)	(%)
OntoNotes 5.0 - English	59,924	81,828	8,528	11,066	8,262	11,057	98.5	41.1
OntoNotes 5.0 - Chinese	36,487	62,543	6,083	9,104	4,472	7,494	92.9	49.1
SemEval2010T1 - Catalan	8,709	15,278	1,445	2,431	1,698	2,910	100.0	28.6
SemEval2010T1 - Spanish	9,022	17,297	1,419	2,615	1,705	3,046	100.0	29.8

Table 2: Dataset statistics. “ST” is the ratio of entities that form subtrees. “GD” is the ratio of entities that have grandchild dependencies within their subtrees.

trees into the Stanford dependency (De Marneffe and Manning, 2008) trees using the rule-based tool (De Marneffe et al., 2006) by Stanford CoreNLP (Manning et al., 2014). For English, Pradhan et al. (2013) provided the train/dev/test split⁴ and the split has been used by several previous works (Chiu and Nichols, 2016; Li et al., 2017; Ghaddar and Langlais, 2018). For Chinese, we use the official splits provided by Pradhan et al. (2012)⁵.

Besides, we also conduct experiments on the Catalan and Spanish datasets from the SemEval-2010 Task 1⁶ (Recasens et al., 2010)⁷. The SemEval-2010 task was originally designed for the task of coreference resolution in multiple languages. Again, we chose these corpora primarily because they contain both dependency and named entity annotations. Following Finkel and Manning (2009) and Jie et al. (2017), we select the most dominant three entity types and merge the rest into one general a entity type “misc”. Table 2 shows the statistics of the datasets used in main experiments. To further evaluate the effectiveness of the dependency structures, we also conduct additional experiments under a low-resource setting for NER (Cotterell and Duh, 2017).

The last two columns of Table 2 show the relationships between the dependency trees and named entities with length larger than 2 for the complete dataset. Specifically, the penultimate column shows the percentage of entities that can form a complete subtree (ST) under their dependency tree structures. Apparently, most of the entities form subtrees, especially for the Catalan and Spanish datasets where 100% entities form subtrees. This observation is consistent with the findings reported in Jie et al. (2017). The last column in Table 2 shows the percentage of the grandchild

⁴<http://cemantix.org/data/ontonotes.html>

⁵<http://conll.cemantix.org/2012/data.html>

⁶<http://stel.ub.edu/semeval2010-coref/download>

⁷This dataset also has English portion but it is a subset of the OntoNotes English.

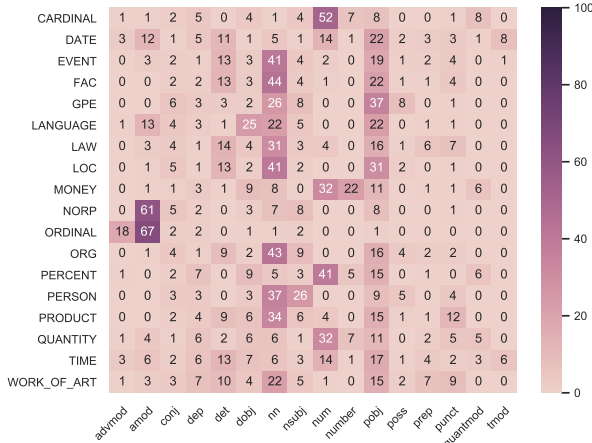


Figure 3: Percentage of entity words (y axis) with respect to dependency relations (x axis) in the OntoNotes English dataset. Columns with percentage less than 5% are ignored for brevity.

dependencies (Koo and Collins, 2010) (GD) that exist in these subtrees (i.e., entities). Such grand-child dependencies could be useful for detecting certain named entities, especially for long entities. As we will see later in Section 5, the performance of long entities can be significantly improved with our dependency-guide model.

The heatmap table in Figure 3 shows the correlation between the entity types and the dependency relations in the OntoNotes English dataset. Specifically, each entry denotes the percentage of the entities that have a parent dependency with a specific dependency relation. For example, at the row with GPE entity, 37% of the entity words⁸ have a dependency edge whose label is “*pobj*”. When looking at column of “*pobj*” and “*nn*”, we can see that most of the entities relate to the prepositional object (*pobj*) and noun compound modifier (*nn*) dependencies. Especially for the NORP (i.e., *nationalities or religious or political groups*) and ORDINAL (e.g., “*first*”, “*second*”) entities, more than 60% of the entity words have the dependency with adjectival modifier (*amod*) relation. Furthermore, every entity type (i.e., row) has a most related dependency relation (with more than 17% occurrences). Such observations present useful information that can be used to categorize named entities with different types.

Baselines We implemented the state-of-the-art NER model **BiLSTM-CRF** (Lample et al., 2016) as the first baseline with different number of LSTM layers ($L = \{0, 1, 2, 3\}$). $L = 0$ indi-

⁸The words that are annotated with entity labels.

cates the model only relies on the input representation. Following Zhang et al. (2018), the complete dependency trees are considered bidirectional and encoded with a contextualized GCN (BiLSTM-GCN). We further add the relation-specific parameters (Marcheggiani and Titov, 2017) and a CRF layer for the NER task. The resulting baseline is **BiLSTM-GCN-CRF**⁹. We use the bootstrapping paired t-test (Berg-Kirkpatrick et al., 2012) for significance test when comparing the results of different models.

Experimental Setup We choose MLP as the interaction function in our DGLSTM-CRF according to performance on the development set. The hidden size of all models (i.e., LSTM, GCN) is set to 200. We use the Glove (Pennington et al., 2014) 100-d word embeddings, which was shown to be effective in English NER task (Ma and Hovy, 2016; Peters et al., 2018a). We use the publicly available FastText (Grave et al., 2018) word embeddings for Chinese, Catalan and Spanish. The ELMo (Peters et al., 2018a), deep contextualized word representations¹⁰ are used for all languages in our experiments since Che et al. (2018) provides ELMo for many other languages¹¹, including Chinese, Catalan and Spanish. We use the average weights over all layers of the ELMo representations and concatenate them with the input representation u . Our models are optimized by mini-batch stochastic gradient descent (SGD) with learning rate 0.01 and batch size 10. The L_2 regularization parameter is $1e-8$. The hyperparameters are selected according to the performance on the OntoNotes English development set.

4.1 Main Results

OntoNotes English Table 3 shows the performance comparison between our work and previous work on the OntoNotes English dataset. Without the LSTM layers (i.e., $L = 0$), the proposed model with dependency information significantly improves the NER performance with more than 2 points in F_1 compared to the baseline BiLSTM-CRF ($L = 0$), which demonstrate the effective-

⁹Detailed description of this baseline can also be found in the supplementary material.

¹⁰We also tried BERT (Devlin et al., 2019) in preliminary experiments and obtained similar performance as ELMo. The NER performance using BERT without fine-tuning reported in Peters et al. (2019) is consistent with the one reported by ELMo (Peters et al., 2018a).

¹¹<https://github.com/HIT-SCIR/ELMoForManyLangs>

Model	Prec.	Rec.	F ₁
Chiu and Nichols (2016)	86.04	86.53	86.28
Li et al. (2017)	88.00	86.50	87.21
Ghaddar and Langlais (2018)	-	-	87.95
Strubell et al. (2017)	-	-	86.84
BiLSTM-CRF ($L = 0$)	82.03	80.78	81.40
BiLSTM-CRF ($L = 1$)	87.21	86.93	87.07
BiLSTM-CRF ($L = 2$)	87.89	87.68	87.78
BiLSTM-CRF ($L = 3$)	87.81	87.50	87.65
BiLSTM-GCN-CRF	88.30	88.06	88.18
DGLSTM-CRF ($L = 0$)	85.31	82.19	84.09
DGLSTM-CRF ($L = 1$)	88.78	87.29	88.03
DGLSTM-CRF ($L = 2$)	88.53	88.50	88.52
DGLSTM-CRF ($L = 3$)	87.59	88.93	88.25
Contextualized Word Representation			
Akbik et al. (2018) (Flair)	-	-	89.30
BiLSTM-CRF ($L = 0$) + ELMo	85.44	84.41	84.92
BiLSTM-CRF ($L = 1$) + ELMo	89.14	88.59	88.87
BiLSTM-CRF ($L = 2$) + ELMo	88.25	89.71	88.98
BiLSTM-CRF ($L = 3$) + ELMo	88.03	89.04	88.53
BiLSTM-GCN-CRF + ELMo	89.40	89.71	89.55
DGLSTM-CRF ($L = 0$) + ELMo	86.87	85.12	85.99
DGLSTM-CRF ($L = 1$) + ELMo	89.40	89.96	89.68
DGLSTM-CRF ($L = 2$) + ELMo	89.59	90.17	89.88
DGLSTM-CRF ($L = 3$) + ELMo	89.43	90.15	89.79

Table 3: Performance comparison on the OntoNotes 5.0 English dataset.

ness of dependencies for the NER task. Our best performing BiLSTM-CRF baseline (with Glove) achieves a F_1 score of 87.78 which is better than or on par with previous works (Chiu and Nichols, 2016; Li et al., 2017; Ghaddar and Langlais, 2018) with extra features. This baseline also outperforms the CNN-based models (Strubell et al., 2017; Li et al., 2017). The BiLSTM-GCN-CRF model outperforms the BiLSTM-CRF model but achieves inferior performance compared to the proposed DGLSTM-CRF model. We believe it is challenging to preserve the surrounding context information with stacking GCN layers while contextual information is important for NER (Peters et al., 2018b). Overall, the 2-layer DGLSTM-CRF model significantly (with $p < 0.01$) outperforms the best BiLSTM-CRF baseline and the BiLSTM-GCN-CRF model. As we can see from the table, increasing the number of layers (e.g., $L = 3$) does not give us further improvements for both BiLSTM-CRF and DGLSTM-CRF because such third-order information (e.g., the relationship among a words parent, its grandparent, and great-grandparent) does not play an important role in indicating the presence of named entities.

Model	Prec.	Rec.	F ₁
Pradhan et al. (2013)	78.20	66.45	71.85
Lattice LSTM (Z&Y, 2018)	76.34	77.01	76.67
BiLSTM-CRF ($L = 0$)	76.67	67.79	71.95
BiLSTM-CRF ($L = 1$)	78.45	74.59	76.47
BiLSTM-CRF ($L = 2$)	77.94	75.33	76.61
BiLSTM-CRF ($L = 3$)	76.17	75.23	75.70
BiLSTM-GCN-CRF	76.35	75.89	76.12
DGLSTM-CRF ($L = 0$)	76.91	70.65	73.65
DGLSTM-CRF ($L = 1$)	77.79	75.29	76.52
DGLSTM-CRF ($L = 2$)	77.40	77.41	77.40
DGLSTM-CRF ($L = 3$)	77.01	74.90	75.94
Contextualized Word Representation			
BiLSTM-CRF ($L = 0$) + ELMo	75.20	73.39	74.28
BiLSTM-CRF ($L = 1$) + ELMo	79.20	79.21	79.20
BiLSTM-CRF ($L = 2$) + ELMo	78.49	79.44	78.96
BiLSTM-CRF ($L = 3$) + ELMo	78.54	79.76	79.14
BiLSTM-GCN-CRF + ELMo	78.71	79.29	79.00
DGLSTM-CRF ($L = 0$) + ELMo	76.27	74.61	75.43
DGLSTM-CRF ($L = 1$) + ELMo	78.91	80.22	79.56
DGLSTM-CRF ($L = 2$) + ELMo	78.86	81.00	79.92
DGLSTM-CRF ($L = 3$) + ELMo	79.30	79.86	79.58

Table 4: Performance comparison on the OntoNotes 5.0 Chinese Dataset.

We further compare the performance of all models with ELMo (Peters et al., 2018a) representations to investigate whether the effect of dependency would be diminished by the contextualized word representations. With $L = 0$, the ELMo representations largely improve the performance of BiLSTM-CRF compared to the BiLSTM-CRF model with word embeddings only but is still 1 point below our DGLSTM-CRF model. The 2-layer DGLSTM-CRF model outperforms the best BiLSTM-CRF baseline with 0.9 points in F_1 ($p < 0.001$). Empirically, we found that among the entities that are correctly predicted by DGLSTM-CRF but wrongly predicted by BiLSTM-CRF, 47% of them are with length more than 2. Our finding shows the 2-layer DGLSTM-CRF model is able to accurately recognize long entities, which can lead to a higher precision. In addition, 51.9% of the entities that are correctly retrieved by DGLSTM-CRF have the dependency relations “*pobj*”, “*nn*” and “*nsubj*”, which have strong correlations with certain named entity types (Figure 3). Such a result demonstrates the effectiveness of dependency relations in improving the recall of NER.

OntoNotes Chinese Table 4 shows the performance comparison on the Chinese datasets. We compare our models against the state-of-the-art

Model	Catalan			Spanish		
	Prec.	Rec.	F ₁	Prec.	Rec.	F ₁
BiLSTM-CRF ($L = 0$)	65.91	49.90	56.80	65.97	52.63	58.55
BiLSTM-CRF ($L = 1$)	76.83	63.47	69.51	78.33	69.89	73.87
BiLSTM-CRF ($L = 2$)	73.79	67.63	70.58	77.73	70.91	74.16
BiLSTM-CRF ($L = 3$)	74.75	67.35	70.86	76.41	72.95	74.64
BiLSTM-GCN-CRF	81.25	75.22	78.12	84.10	79.88	81.93

DGLSTM-CRF ($L = 0$)	73.42	61.79	67.10	74.90	61.21	67.38
DGLSTM-CRF ($L = 1$)	81.87	79.28	80.55	83.21	81.19	82.19
DGLSTM-CRF ($L = 2$)	83.35	80.00	81.64	84.05	82.90	83.47
DGLSTM-CRF ($L = 3$)	81.87	80.21	81.03	84.12	83.45	83.78
Contextualized Word Representation						
BiLSTM-CRF ($L = 0$) + ELMo	67.53	64.47	65.96	73.16	69.01	71.03
BiLSTM-CRF ($L = 1$) + ELMo	77.85	76.22	77.03	81.72	79.09	80.38
BiLSTM-CRF ($L = 2$) + ELMo	78.61	78.32	78.46	80.89	80.30	80.59
BiLSTM-CRF ($L = 3$) + ELMo	79.11	77.32	78.21	80.48	79.45	79.96
BiLSTM-GCN-CRF + ELMo	83.68	83.16	83.42	85.31	85.19	85.25

DGLSTM-CRF ($L = 0$) + ELMo	70.87	65.81	68.25	75.96	72.52	74.20
DGLSTM-CRF ($L = 1$) + ELMo	82.29	82.37	82.33	84.05	84.77	84.41
DGLSTM-CRF ($L = 2$) + ELMo	84.71	83.75	84.22	87.79	87.33	87.56
DGLSTM-CRF ($L = 3$) + ELMo	84.50	83.92	84.21	86.74	86.57	86.66

Table 5: Results on the SemEval-2010 Task 1 datasets.

NER model on this dataset, Lattice LSTM (Zhang and Yang, 2018)¹². Our implementation of the strong BiLSTM-CRF baseline achieves comparable performance against the Lattice LSTM. Similar to the English dataset, our model with $L = 0$ significantly improves the performance compared to the BiLSTM-CRF ($L = 0$) model. Our DGLSTM-CRF model achieves the best performance with $L = 2$ and is consistently better ($p < 0.02$) than the strong BiLSTM-CRF baselines. As we can see from the table, the improvements of the DGLSTM-CRF model mainly come from recall ($p < 0.001$) compared to the BiLSTM model, especially in the scenario with word embeddings only. Empirically, we also found that those correctly retrieved entities of the DGLSTM-CRF (compared against the baseline) mostly correlate with the following dependency relations: “*nn*”, “*nsubj*”, “*nummod*”. However, DGLSTM-CRF achieves lower precisions against BiLSTM-CRF, which indicates that the DGLSTM-CRF model makes more false-positive predictions. The reason could be the relatively lower ratio of ST(%)¹³ as shown in Table 2, which means some of the entities do not form subtrees under the complete dependency trees. In such a scenario, the model may not correctly identify the boundary of the entities, which results in lower precision.

¹²We run their code on the OntoNotes 5.0 Chinese dataset.

¹³Percentage of entities that can form a subtree.

SemEval-2010 Table 5 shows the results of our models on the SemEval-2010 Task 1 datasets. Overall, we observe substantial improvements of the DGLSTM-CRF on the Catalan and Spanish datasets (with $p < 0.001$ marked in bold against the best performing BiLSTM-CRF baseline), especially for DGLSTM-CRF with ELMo and L larger than 1. With word embeddings, the best DGLSTM-CRF model outperforms the best performing BiLSTM-CRF baseline with more than 10 and 9 points in F_1 on the Catalan and Spanish datasets, respectively. The BiLSTM-GCN-CRF model also performs much better than the BiLSTM-CRF baselines but is worse than the DGLSTM-CRF model with $L \geq 2$. Both precision and recall significantly improve with a large margin compared to the best performing BiLSTM-CRF, especially for the recall (with more than 10 points improvement) on these two datasets. With ELMo, the best performing DGLSTM-CRF model outperforms the BiLSTM-CRF baseline with about 6 and 7 points in F_1 on these two datasets, respectively. The substantial improvements show that the structural dependency information is extremely helpful for these two datasets.

With ELMo representations, we observe about 2 and 3 points improvements in F_1 compared with the 1-layer DGLSTM-CRF model on these two datasets, respectively. Empirically, more than 50% of the entities that are correctly predicted by the

Model	Prec.	Rec.	F ₁
Peters et al. (2018a) ELMo	-	-	92.2
BiLSTM-CRF + ELMo ($L = 2$)	92.1	92.3	92.2
DGLSTM-CRF + ELMo ($L = 2$)	92.2	92.5	92.4

Table 6: Performance on the CoNLL-2003 English dataset.

Model	Catalan			Spanish		
	Prec.	Rec.	F ₁	Prec.	Rec.	F ₁
BiLSTM-CRF ($L = 1$)	47.88	18.59	26.78	40.77	19.01	25.93
DGLSTM-CRF ($L = 1$)	47.71	31.55	37.98	49.39	31.91	38.77
- with gold dependency	52.13	33.26	40.61	52.14	35.59	42.30

Table 7: Low-resource NER performance on the SemEval-2010 Task 1 datasets.

2-layer model but not the 1-layer model are with length larger than 2. Also, most of these entities contain the grandchild dependencies “(sn, sn)” and “(spec, sn)” where *sn* represents noun phrase and *spec* represents specifier (e.g., determiner, quantifier) in both datasets. Such a finding shows that the 2-layer model is able to capture the interactions given by the grandchild dependencies.

4.2 Additional Experiments

CoNLL-2003 English Table 6 shows the performance on the CoNLL-2003 English dataset. The dependencies are predicted from Spacy (Honni-bal and Montani, 2017). With the contextualized word representations, DGLSTM-CRF outperforms BiLSTM-CRF with 0.2 points in F₁ ($p < 0.09$). The improvement is not significant due to the relatively lower equality of the dependency trees. To further study the effect of the dependencies, we modified the predicted dependencies to ensure each entity form a subtree in the complete dataset. Such modification improves the F₁ to 92.7, which is significantly better ($p < 0.05$) than the BiLSTM-CRF.

Low-Resource NER Following Cotterell and Duh (2017), we emulate truly low-resource condition with 100 sentences for training. We assume that the contextualized word representations are not available and dependencies are predicted. Table 7 shows the NER performance on the SemEval-2010 Task 1 datasets under the low-resource setting. With limited amount of training data, BiLSTM-CRF suffers from low recall and the DGLSTM-CRF largely improves it on these two datasets. Using gold dependencies further significantly improves the precision and recall.

	English	Chinese	Catalan	Spanish
BiLSTM-CRF	88.98	79.20	78.46	80.59
(Dependency LAS) [†]	(94.89)	(89.28)	(93.25)	(93.35)
DGLSTM-CRF (Predicted)	89.64	79.59	82.37	83.92
Improvement Δ	+0.66	+0.39	+3.91	+3.33
DGLSTM-CRF (Gold)	89.88	79.92	84.22	87.56

Table 8: F₁ performance of DGLSTM-CRF with predicted dependencies against the best performing BiLSTM-CRF. [†]: LAS is label attachment score which is the metric for dependency evaluation.

Model	Prec.	Rec.	F ₁
BiLSTM-CRF + ELMo ($L = 2$)	89.14	88.59	88.87
DGLSTM-CRF + ELMo ($L = 2$)	89.59	90.17	89.88
- $g(\cdot) =$ self connection	89.17	90.08	89.62
- $g(\cdot) =$ Concatenation	89.43	90.09	89.76
- $g(\cdot) =$ Addition	89.24	89.78	89.50
-w/o dependency relation	88.92	89.99	89.46

Table 9: Ablation study of the DGLSTM-CRF model on the OntoNotes English dataset.

Effect of Dependency Quality To evaluate how the quality of dependency trees affect the performance, we train a state-of-the-art dependency parser (Dozat and Manning, 2017) using our training set and make prediction on the development/test set. We implemented the dependency parser using the AllenNLP package (Gardner et al., 2017). Table 8 shows the performance (LAS) of the dependency parser on four languages (i.e., OntoNotes English, OntoNotes Chinese, Catalan and Spanish) and the performance of DGLSTM-CRF against the best performing BiLSTM-CRF with ELMo. DGLSTM-CRF even with predicted dependencies is able to consistently outperform the BiLSTM-CRF on four languages. However, the performance is still worse than the DGLSTM-CRF with gold dependencies, especially on the Catalan and Spanish. Such results suggest that it is essential to have high-quality dependency annotations available for the proposed model.

Ablation Study Table 9 shows the ablation study of the 2-layer DGLSTM-CRF model on the OntoNotes English dataset. With self connection as interaction function, the F₁ drops 0.3 points. The model achieves comparable performance with concatenation as interaction function but F₁ drops about 0.4 points with the addition interaction function. We believe that the addition potentially leads to certain information loss. Without the depen-

Dataset	Model	Entity Length					
		1	2	3	4	5	≥6
English	BiLSTM-CRF	91.8	88.5	83.4	84.0	75.4	76.0
	DGLSTM-CRF	91.8	90.1	85.4	87.0	80.8	78.7
Chinese	BiLSTM-CRF	81.2	74.3	73.1	62.8	70.3	57.5
	DGLSTM-CRF	82.2	75.5	71.8	64.1	58.5	41.1
Catalan	BiLSTM-CRF	80.5	81.0	75.8	56.1	45.0	38.4
	DGLSTM-CRF	85.4	85.1	84.1	78.9	60.9	59.3
Spanish	BiLSTM-CRF	84.2	81.1	81.0	53.3	53.3	37.1
	DGLSTM-CRF	89.3	87.4	90.8	74.1	67.7	64.4

Table 10: Performance of entities with different lengths on the four datasets: OntoNotes (English), OntoNotes Chinese, Catalan and Spanish.

dependency relation embedding \mathbf{v}_r in the input representation, the F_1 drops about 0.4 points.

5 Analysis

5.1 Effectiveness of Dependency Relations

To demonstrate whether the model benefits from the dependency relations, we first select the entities that are correctly predicted by the 2-layer DGLSTM-CRF model but not by the best performing baseline 2-layer BiLSTM-CRF on the OntoNotes English dataset. We draw the heatmap in Figure 4 based on these entities. Comparing Figure 3 and 4, we can see that they are similar in terms of the density. Both of them show consistent relationships between the entity types and the dependency relations. The comparison shows that the improvements partially result from the effect of dependency relations. We also found from our model’s predictions that some entity types have strong correlations with the relation pairs on grandchild dependencies¹⁴.

5.2 Entity with Different Lengths

Table 10 shows the performance comparison with different entity lengths on all datasets. As mentioned earlier, the dependencies as well as the grandchild relations allow our models to capture the long-distance interactions between the words. As shown in the table, the performance of entities with lengths more than 1 consistently improves with DGLSTM-CRF for all languages except Chinese. As we pointed out in the dataset statistics (Table 2), the number of entities that form subtrees in OntoNotes Chinese is relatively smaller compared to other datasets. The performance gain is more significant for entities with longer length on

¹⁴The corresponding heatmap visualization is provided in supplementary material.

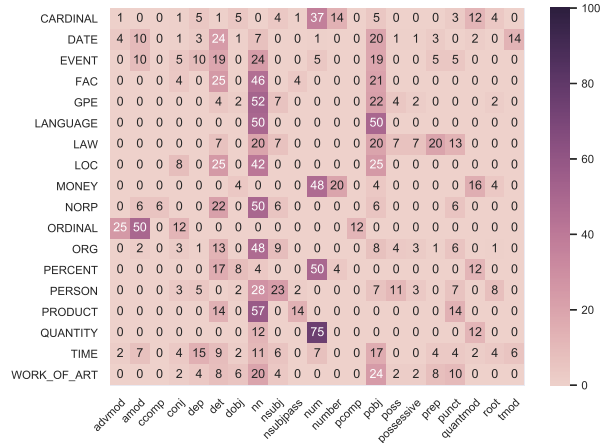


Figure 4: Correlations between the correctly predicted entities and the dependency relations.

the other three languages. We found that, among the improvements of entities with length larger than 2 in English, 85% of them have long-distance dependencies and 30% of them have grandchild dependencies within the entity boundary. The analysis shows that our model that exploits the dependency tree structures is helpful for recognizing long entities.

6 Conclusions and Future Work

Motivated by the relationships between the dependency trees and named entities, we propose a dependency-guided LSTM-CRF model to encode the complete dependency tree and capture such relationships for the NER task. Through extensive experiments on several datasets, we demonstrate the effectiveness of the proposed model in improving the NER performance. Our analysis shows that NER benefits from the dependency relations and long-distance dependencies, which are able to capture the non-local interactions between the words.

As statistics shows that most of the entities form subtrees under the dependency trees, future work includes building a model for joint NER and dependency parsing which regards each entity as a single unit in a dependency tree.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments on this work. We would also like to thank Zhijiang Guo and Yan Zhang for the fruitful discussion. This work is supported by Singapore Ministry of Education Academic Research Fund (AcRF) Tier 2 Project MOE2017-T2-1-156.

References

- Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. [Contextual string embeddings for sequence labeling](#). In *Proceedings of COLING*.
- Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. [An empirical investigation of statistical significance in nlp](#). In *Proceedings of EMNLP-CoNLL*.
- Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. [Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation](#). In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*.
- Jason P. C. Chiu and Eric Nichols. 2016. [Named entity recognition with bidirectional lstm-cnns](#). *Transactions of the Association of Computational Linguistics*, 4:357–370.
- Corinna Cortes and Vladimir Vapnik. 1995. [Support-vector networks](#). *Machine learning*, 20(3):273–297.
- Ryan Cotterell and Kevin Duh. 2017. [Low-resource named entity recognition with cross-lingual, character-level neural conditional random fields](#). In *Proceedings of IJCNLP*.
- Alessandro Cucchiarelli and Paola Velardi. 2001. [Un-supervised named entity recognition using syntactic and semantic contextual evidence](#). *Computational Linguistics*, 27(1):123–131.
- Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. [Generating typed dependency parses from phrase structure parses](#). In *Proceedings of LREC*.
- Marie-Catherine De Marneffe and Christopher D Manning. 2008. [Stanford typed dependencies manual](#). 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 NAACL*.
- Li Dong and Mirella Lapata. 2018. [Coarse-to-fine decoding for neural semantic parsing](#). In *Proceedings of ACL*.
- Timothy Dozat and Christopher D Manning. 2017. [Deep biaffine attention for neural dependency parsing](#). In *Proceedings of ICLR*.
- Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. [Incorporating non-local information into information extraction systems by gibbs sampling](#). In *Proceedings of ACL*.
- Jenny Rose Finkel and Christopher D Manning. 2009. [Joint parsing and named entity recognition](#). In *Proceedings of NAACL*.
- Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. [Allennlp: A deep semantic natural language processing platform](#). In *Proceedings of Workshop for NLP-OSS*.
- Abbas Ghaddar and Phillippe Langlais. 2018. [Robust lexical features for improved neural network named-entity recognition](#). In *Proceedings of COLING*.
- Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. [Learning word vectors for 157 languages](#). In *Proceedings of LREC*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long short-term memory](#). *Neural computation*, 9(8):1735–1780.
- Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. To appear.
- Zhanming Jie, Aldrian Obaja Muis, and Wei Lu. 2017. [Efficient dependency-guided named entity recognition](#). In *Proceedings of AAAI*.
- Thomas N Kipf and Max Welling. 2017. [Semi-supervised classification with graph convolutional networks](#). In *Proceedings of ICLR*.
- Terry Koo and Michael Collins. 2010. [Efficient third-order dependency parsers](#). In *Proceedings of ACL*.
- John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. [Conditional random fields: Probabilistic models for segmenting and labeling sequence data](#). In *Proceedings of ICML*.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. [Neural architectures for named entity recognition](#). In *Proceedings of NAACL-HLT*.
- Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. [End-to-end neural coreference resolution](#). In *Proceedings of EMNLP*.
- 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 EMNLP*.
- Xiao Ling and Daniel S Weld. 2012. [Fine-grained entity recognition](#). In *Proceedings of AAAI*.
- Jingchen Liu, Minlie Huang, and Xiaoyan Zhu. 2010. [Recognizing biomedical named entities using skip-chain conditional random fields](#). In *Proceedings of the Workshop on BioNLP*.
- Xuezhe Ma and Eduard Hovy. 2016. [End-to-end sequence labeling via bi-directional lstm-cnns-crf](#). In *Proceedings of ACL*.

- Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. [The stanford corenlp natural language processing toolkit](#). In *Proceedings of ACL: System Demonstrations*.
- Diego Marcheggiani and Ivan Titov. 2017. [Encoding sentences with graph convolutional networks for semantic role labeling](#). In *Proceedings of EMNLP*.
- Makoto Miwa and Mohit Bansal. 2016. [End-to-end relation extraction using lstms on sequences and tree structures](#). In *Proceedings of ACL*.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [Glove: Global vectors for word representation](#). In *Proceedings of EMNLP*.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. [Deep contextualized word representations](#). In *Proceedings of NAACL*.
- Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. [Dissecting contextual word embeddings: Architecture and representation](#). In *Proceedings of EMNLP*.
- Matthew Peters, Sebastian Ruder, and Noah A Smith. 2019. [To tune or not to tune? adapting pretrained representations to diverse tasks](#). *arXiv preprint arXiv:1903.05987*.
- 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 CoNLL*.
- Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. [Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes](#). In *Proceedings of Joint EMNLP-CoNLL: Shared Task*.
- Lance A Ramshaw and Mitchell P Marcus. 1999. [Text chunking using transformation-based learning](#). In *Proceedings of Third Workshop on Very Large Corpora*.
- Lev Ratinov and Dan Roth. 2009. [Design challenges and misconceptions in named entity recognition](#). In *Proceedings of CoNLL*.
- 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*.
- Nils Reimers and Iryna Gurevych. 2017. [Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging](#). In *Proceedings of EMNLP*.
- Sunita Sarawagi and William W Cohen. 2004. [Semi-markov conditional random fields for information extraction](#). In *Proceedings of NeurIPS*.
- Ryohei Sasano and Sadao Kurohashi. 2008. [Japanese named entity recognition using structural natural language processing](#). In *Proceedings of IJCNLP*.
- Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. [Fast and accurate entity recognition with iterated dilated convolutions](#). In *Proceedings of EMNLP*.
- Charles Sutton and Andrew McCallum. 2004. [Collective segmentation and labeling of distant entities in information extraction](#). In *Proceedings of the ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*.
- 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*.
- Yue Zhang and Jie Yang. 2018. [Chinese ner using lattice lstm](#). In *Proceedings of ACL*.
- Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. [Graph convolution over pruned dependency trees improves relation extraction](#). In *Proceedings of EMNLP*.