Better Feature Integration for Named Entity Recognition

Lu Xu[1,2]*, **Zhanming Jie**[1,3], **Wei Lu**[1], **Lidong Bing**[2]
[1] StatNLP Research Group, Singapore University of Technology and Design
[2] DAMO Academy, Alibaba Group [3] ByteDance
xu_lu@mymail.sutd.edu.sg, allan@bytedance.com
luwei@sutd.edu.sg, l.bing@alibaba-inc.com

Abstract

It has been shown that named entity recognition (NER) could benefit from incorporating the long-distance structured information captured by dependency trees. We believe this is because both types of features – the contextual information captured by the linear sequences and the structured information captured by the dependency trees may complement each other. However, existing approaches largely focused on stacking the LSTM and graph neural networks such as graph convolutional networks (GCNs) for building improved NER models, where the exact interaction mechanism between the two different types of features is not very clear, and the performance gain does not appear to be significant. In this work, we propose a simple and robust solution to incorporate both types of features with our Synergized-LSTM (Syn-LSTM), which clearly captures how the two types of features interact. We conduct extensive experiments on several standard datasets across four languages. The results demonstrate that the proposed model achieves better performance than previous approaches while requiring fewer parameters. Our further analysis demonstrates that our model can capture longer dependencies compared with strong baselines.[1]

1 Introduction

Named entity recognition (NER) is one of the most fundamental and important tasks in natural language processing (NLP). While the literature (Peters et al., 2018; Akbik et al., 2018; Devlin et al., 2019) largely focuses on training deep language models to improve the contextualized word representations, previous studies show that

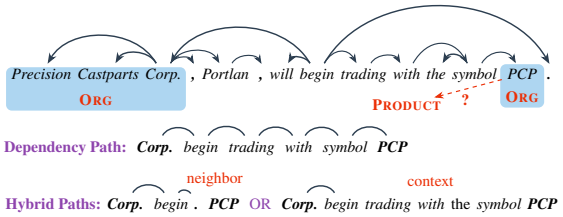

Figure 1: A sentence annotated with dependency trees and named entities. The paths to connect two entities are shown below the sentence.

the structured information such as interactions between non-adjacent words can also be important for NER (Finkel et al., 2005; Jie et al., 2017; Aguilar and Solorio, 2019).

However, sequence models such as bidirectional LSTM (Hochreiter and Schmidhuber, 1997) are not able to fully capture the long-range dependencies (Bengio, 2009). For instance, Figure 1 (top) shows one type of structured information in NER. The words "*Precision Castparts Corp.*" can be easily inferred as ORGANIZATION by its context (i.e., *Corp.*). However, the second entity "*PCP*" could be misclassified as a PRODUCT entity if a model relies more on the context "*begin trading with*" but ignores the hidden information that "*PCP*" is the symbol of "*Precision Castparts Corp.*".

Previous research works (Li et al., 2017; Jie and Lu, 2019; Wang et al., 2019) have been using the parse trees (Chomsky, 1956, 1969; Sandra and Taft, 2014) to incorporate such structured information. Figure 1 (Dependency Path) shows that the first entity can be connected to the second entity following the dependency tree with 5 hops. Incorporating the dependency information can be done with graph neural networks (GNNs) such as graph convolutional networks (GCNs) (Kipf and Welling, 2017). However, simply stacking the LSTM and GCN architectures for NER can only provide us with modest improvements; sometimes, it decreases performance (Jie and Lu, 2019). Based on the depen-

∗ Lu Xu is under the Joint PhD Program between Alibaba and Singapore University of Technology and Design. The work was done when Zhanming Jie was a PhD student in Singapore University of Technology and Design.

[1]We make our code publicly available at https://github.com/xuuluuu/SynLSTM-for-NER.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3457–3469
June 6–11, 2021.

dency path in Figure 1, it requires a 5-layer GCN to capture the connections between these two entities. However, deep GCN architectures often face training difficulties, which cause a performance drop (Hamilton et al., 2017b; Kipf and Welling, 2017). Directly stacking GCN and LSTM has difficulties in modeling the interaction between dependency trees and contextual information.

To address the above limitations, we propose the Synergized-LSTM (Syn-LSTM), a new recurrent neural network architecture that considers an additional graph-encoded representation to update the memory and hidden states, as shown in Figure 2. More specifically, the graph-encoded representation for each word can be obtained with GCNs. Our proposed Syn-LSTM allows the cell to receive the structured information from the graph-encoded representation. With the newly designed gating mechanism, our model is able to make independent assessments on the amounts of information to be retrieved from the word representation and the graph-encoded representation respectively. Such a mechanism allows for better integration of both contextual and structured information.

Our contributions can be summarized as:

- We propose a simple and robust Syn-LSTM model to better incorporate the structured information conveyed by dependency trees. The output of the Syn-LSTM cell is jointly determined by both contextual and structured information. We adopt the classic conditional random fields (CRF) (Lafferty et al., 2001) on top of the Syn-LSTM for NER.
- We conduct extensive experiments on several standard datasets across four languages. The proposed model significantly outperforms previous approaches on these datasets.
- We show that the proposed model can capture long-distance interactions between entities. Our further analysis statistically demonstrates the proposed gating mechanism is able to aggregate the structured information selectively.

2 Synergized-LSTM

2.1 Incorporating Structured Information

To incorporate the long-range dependencies, we consider an additional graph-encoded representation $\mathbf{g}_t$ (Figure 2) as the model input to integrate

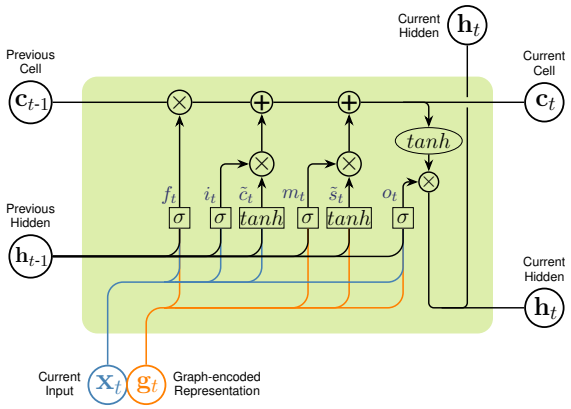

Figure 2: Syn-LSTM cell. t is the current time step.

contextual and structured information. The graph-encoded representation $\mathbf{g}_t$ can be derived from Graph Neural Networks (GNNs) such as GCN (Kipf and Welling, 2017), which are capable of bringing in structured information through graph structure (Hamilton et al., 2017a).

However, structured information sometimes is hard to encode, as we can see from the example in Figure 1. One naive approach is to use a deep GNN to capture such information along multiple dependency arcs between two words, which could mess up information and lead to training difficulties. A straightforward solution is to integrate both structured and contextual information via LSTM. As shown in Figure 1 (Hybrid Paths), the structured information can be passed to neighbors or context, which allows a model to use less number of GNN layers and alleviate such issues for long-range dependencies. The input to the LSTM can simply be the concatenation of word representation $\mathbf{x}_t$ and $\mathbf{g}_t$ at each position (Jie and Lu, 2019)[2]. However, because such an approach requires both $\mathbf{x}_t$ and $\mathbf{g}_t$ to decide the value of the input gate jointly, it could be a potential victim of two sources of uncertainties: 1) the uncertainty of the quality of graph-encoded representation $\mathbf{g}_t$, and 2) the uncertainty of the exact interaction mechanism between the two types of features. These may lead to sub-optimal performance, especially if the graph-encoded representation $\mathbf{g}_t$ is unsatisfactory. Thus, we need to design a new approach to incorporate both types of information from $\mathbf{x}_t$ and $\mathbf{g}_t$ with a more explicit interaction mechanism, with which we hope to alleviate the above issues.

[2]They concatenate the current word and head word representations.

2.2 Syn-LSTM Cell

We propose the Synergized-LSTM (Syn-LSTM) to better integrate the contextual and structured information to address the above limitations. The inputs of the Syn-LSTM cell include previous cell state $\mathbf{c}_{t-1}$, previous hidden state $\mathbf{h}_{t-1}$, current cell input $\mathbf{x}_t$, and an additional graph-encoded representation $\mathbf{g}_t$. The outputs of the Syn-LSTM cell include current cell state $\mathbf{c}_t$ and current hidden state $\mathbf{h}_t$. Within the cell, there are four gates: input gate $\mathbf{i}_t$, forget gate $\mathbf{f}_t$, output gate $\mathbf{o}_t$, and an additional new gate $\mathbf{m}_t$ to control the flow of information. Note that the forget gate $\mathbf{f}_t$ and output gate $\mathbf{o}_t$ are not just looking at $\mathbf{h}_{t-1}$ and $\mathbf{x}_t$; they are also affected by the graph-encoded representation $\mathbf{g}_t$. The cell state $\mathbf{c}_t$ and hidden state $\mathbf{h}_t$ are computed as follows:

$$\mathbf{f}_t = \sigma(W^{(f)}\mathbf{x}_t + U^{(f)}\mathbf{h}_{t-1} + Q^{(f)}\mathbf{g}_t + \mathbf{b}^{(f)}) \quad (1)$$

$$\mathbf{o}_t = \sigma(W^{(o)}\mathbf{x}_t + U^{(o)}\mathbf{h}_{t-1} + Q^{(o)}\mathbf{g}_t + \mathbf{b}^{(o)}) \quad (2)$$

$$\mathbf{i}_t = \sigma(W^{(i)}\mathbf{x}_t + U^{(i)}\mathbf{h}_{t-1} + \mathbf{b}^{(i)}) \quad (3)$$

$$\mathbf{m}_t = \sigma(W^{(m)}\mathbf{g}_t + U^{(m)}\mathbf{h}_{t-1} + \mathbf{b}^{(m)}) \quad (4)$$

$$\tilde{\mathbf{c}}_t = \tanh(W^{(u)}\mathbf{x}_t + U^{(u)}\mathbf{h}_{t-1} + \mathbf{b}^{(u)}) \quad (5)$$

$$\tilde{\mathbf{s}}_t = \tanh(W^{(n)}\mathbf{g}_t + U^{(n)}\mathbf{h}_{t-1} + \mathbf{b}^{(n)}) \quad (6)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t + \mathbf{m}_t \odot \tilde{\mathbf{s}}_t \quad (7)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (8)$$

where σ is the sigmoid function, $W^{(\cdot)}$, $U^{(\cdot)}$, $Q^{(\cdot)}$ and $\mathbf{b}^{(\cdot)}$ are learnable parameters.

The additional new gate $\mathbf{m}_t$ is used to control the information from the graph-encoded representation directly. Such a design allows the original input gates $\mathbf{i}_t$ and our new gate $\mathbf{m}_t$ to make independent assessments on the amounts of information to be retrieved from the word representation $\mathbf{x}_t$ and the graph-encoded representation $\mathbf{g}_t$ respectively. On the other hand, we also have a different candidate state $\tilde{\mathbf{s}}_t$ to represent the cell state that corresponds to the graph-encoded representation separately.

With the proposed Syn-LSTM, the structured information captured by the dependency trees can be passed to each cell, and the additional gate $\mathbf{m}_t$ is able to control how much structured information can be incorporated. The additional gate enables the model to feed the contextual and structured information into the LSTM cell separately. Such a mechanism allows our model to aggregate the information from linear sequence and dependency trees selectively.

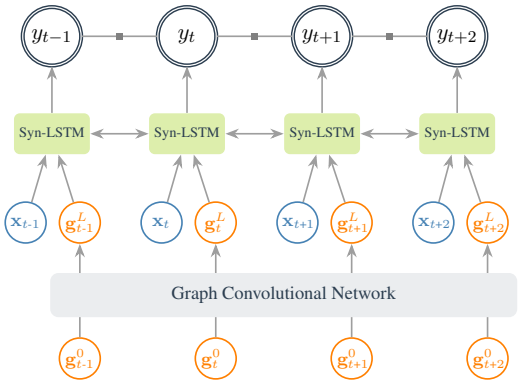

Figure 3: Syn-LSTM-CRF architecture.

Similar to the previous work (Levy et al., 2018), it is also possible to show that the cell state $\mathbf{c}_t$ implicitly computes the element-wise weighted sum of the previous states by expanding Equation 7:

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t + \mathbf{m}_t \odot \tilde{\mathbf{s}}_t \quad (9)$$

$$= \sum_{j=0}^{t}(\mathbf{i}_j \odot \prod_{k=j+1}^{t} \mathbf{f}_k) \odot \tilde{\mathbf{c}}_j + \sum_{j=0}^{t}(\mathbf{m}_j \odot \prod_{k=j+1}^{t} \mathbf{f}_k) \odot \tilde{\mathbf{s}}_j \quad (10)$$

$$= \sum_{j=0}^{t} \mathbf{a}_j^t \odot \tilde{\mathbf{c}}_j + \sum_{j=0}^{t} \mathbf{q}_j^t \odot \tilde{\mathbf{s}}_j \quad (11)$$

Note that the two terms, $\mathbf{a}_j^t$ and $\mathbf{q}_j^t$, are the product of gates. The value of the two terms are in the range from 0 to 1. Since the $\tilde{\mathbf{c}}_t$ and $\tilde{\mathbf{s}}_t$ represent contextual and structured features, the corresponding weights control the flow of information.

3 Syn-LSTM-CRF

The goal of named entity recognition is to predict the label sequence $\mathbf{y} = \{y_1, y_2, ..., y_n\}$ given the input sequence $\mathbf{w} = \{w_1, w_2, ..., w_n\}$, where w_t represents the t-th word and n is the number of words. Our model is mainly constructed with three layers: input representation layer, bi-directional Syn-LSTM layer, and CRF layer. The architecture of our Syn-LSTM-CRF is shown in Figure 3.

Input Representation Layer Similar to the work by Lample et al. (2016), our input representation also includes the character embeddings, which are the hidden states of character-based BiLSTM. Jie and Lu (2019) highlight that the dependency relation helps to enhance the input representation. Furthermore, previous methods (Wang et al., 2018;

Wang and Lu, 2018) use embeddings of part-of-speech (POS) tags as additional input representation. The input representation $\mathbf{x}_t$ of our model is the concatenation of the word embedding $\mathbf{v}_t$, the character representation $\mathbf{e}_t$, the dependency relation embedding $\mathbf{r}_t$, and the POS embedding $\mathbf{p}_t$:

$$\mathbf{x}_t = [\mathbf{v}_t;\ \mathbf{e}_t;\ \mathbf{r}_t;\ \mathbf{p}_t] \tag{12}$$

where both $\mathbf{r}_t$ and $\mathbf{p}_t$ embeddings are randomly initialized and are fine-tuned during training. For experiments with the contextualized representations (e.g., BERT (Devlin et al., 2019)), we further concatenate the contextual word representation to $\mathbf{x}_t$.

For our task, we employ the graph convolutional network (Kipf and Welling, 2017; Zhang et al., 2018b) to get the graph-encoded representation $\mathbf{g}_t$. Given a graph, an adjacency matrix A of size $n \times n$ is able to represent the graph structure, where n is the number of nodes; $A_{i,j} = 1$ indicates that node i and node j are connected. We transform dependency tree into its corresponding adjacency matrix[3] A, and $A_{i,j} = 1$ denotes that node i and node j have dependency relation. Note that the purpose of graph-encoded representation $\mathbf{g}_t$ is to incorporate the dependency information from neighbor nodes. The input and output representations of the l-th layer GCN at t-th position are denoted as $\mathbf{g}_t^{l-1}$ and $\mathbf{g}_t^l$ respectively. Similar to the work by Zhang et al. (2018b), we use $d_t = \sum_{j=1}^{n} A_{t,j}$, which is the total number of neighbors of node t, to normalize the representation before going through the nonlinear function. The GCN operation is defined as:

$$\mathbf{g}_t^l = \mathrm{ReLU}(\sum_{j=1}^{n} A_{t,j} W^l \mathbf{g}_t^{l-1}/d_t + \mathbf{b}^l) \tag{13}$$

where W^l is a linear transformation and $\mathbf{b}^l$ is a bias. The initial $\mathbf{g}_t^0$ is the concatenation of word embedding $\mathbf{v}_t$, character embedding $\mathbf{e}_t$, and dependency relation embedding $\mathbf{r}_t$: $\mathbf{g}_t^0 = [\mathbf{v}_t;\ \mathbf{e}_t;\ \mathbf{r}_t]$.

Bi-directional Syn-LSTM Layer With the word representation $\mathbf{x}_t$ and the graph-encoded representation $\mathbf{g}_t$, a bi-directional Syn-LSTM is applied to generate contextual representation. The forward and backward Syn-LSTM enable the model to integrate the contextual and structured information from both directions. We concatenate the hidden state $\overrightarrow{\mathbf{h}_t}$ from forward Syn-LSTM and hidden state $\overleftarrow{\mathbf{h}_t}$ from backward Syn-LSTM to form the contextual representation of t-th token: $\mathbf{h}_t = [\overrightarrow{\mathbf{h}_t};\ \overleftarrow{\mathbf{h}_t}]$.

[3]We treat the dependency edge as undirected and add a self-loop for each node: $A_{i,j} = A_{j,i}$ and $A_{i,i} = 1$.

Dataset		# Sent.	# Entity in Sentence Length				
			≤ 14	15 - 29	30 - 44	45 - 59	≥ 60
Catalan	Train	8,709	944	4,821	5,309	2,815	1,389
	Dev	1,445	135	836	815	477	168
	Test	1,698	243	919	946	518	284
Spanish	Train	9,022	855	4,031	6,656	4,279	1,446
	Dev	1,419	125	612	911	707	260
	Test	1,705	175	703	1,143	783	242
English	Train	59,924	13,309	33,853	22728	8,099	3,839
	Dev	8,528	1,778	4,830	2,882	1,051	525
	Test	8,262	1,785	4,673	3,171	1,082	546
Chinese	Train	36,487	8,424	21,033	17,260	8,392	7,434
	Dev	6,083	1,493	3,250	2,284	1,099	978
	Test	4,472	968	2,517	2,149	1,024	836

Table 1: Statistics of datasets.

CRF Layer The CRF (Lafferty et al., 2001) is widely used in NER tasks as it is capable of capturing the structured correlations between adjacent output labels. Given the sentence $\mathbf{w}$ and dependency tree τ, the probability of the label sequence $\mathbf{y}$ is defined as:

$$P(\mathbf{y}|\mathbf{w}, \tau) = \frac{\exp(score(\mathbf{w}, \tau, \mathbf{y}))}{\sum_{\mathbf{y}'} \exp(score(\mathbf{w}, \tau, \mathbf{y}'))} \tag{14}$$

The score function is defined as:

$$score(\mathbf{w}, \tau, \mathbf{y}) = \sum_{t=0}^{n} T_{y_t, y_{t+1}} + \sum_{t=1}^{n} E_{y_t} \tag{15}$$

where $T_{y_t, y_{t+1}}$ denotes the transition score from label y_t to y_{t+1}, E_{y_t} denotes the score of label y_t at the t-th position and the scores are computed using the hidden state $\mathbf{h}_t$. We learn the model parameters by minimizing the negative log-likelihood and employ the Viterbi algorithm to obtain the best label sequence during evaluation.

4 Experiments

Datasets The proposed model is evaluated on four benchmark datasets: SemEval 2010 Task 1 (Recasens et al., 2010) Catalan and Spanish datasets, and OntoNotes 5.0 (Weischedel et al., 2013) English and Chinese datasets. We choose these four datasets as they have explicit dependency annotations which allow us to evaluate the effectiveness of our approach when dependency trees of different qualities are used. For SemEval 2010 Task 1 datasets, there are 4 entity types: PER, LOC and ORG and MISC. For OntoNotes 5.0 datasets, there are 18 entity types in total. Following the work

by Jie and Lu (2019), we transform the parse trees into the Stanford dependency trees (De Marneffe and Manning, 2008) by using Stanford CoreNLP (Manning et al., 2014). Detailed statistics of each dataset can be found in Table 1. Intuitively, longer sentences would require the model to capture more long-distance interactions in the sentences. We present the number of entities in terms of different sentence lengths to show that these datasets have a modest amount of entities in long sentences.

Experimental Setup For Catalan, Spanish, and Chinese, we use the FastText (Grave 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. Specifically, we use bert-as-service (Xiao, 2018) to generate the contextualized word representation without fine-tuning. Following Luo et al. (2020), we use the cased version of BERT large model for the experiments on the OntoNotes 5.0 English data. We use the cased version of BERT base model for the experiments on the other three datasets. For the character embedding, we randomly initialize the character embeddings and set the dimension as 30, and set the hidden size of character-level BiLSTM as 50. The hidden size of GCN and Syn-LSTM is set as 200, the number of GCN layer is 2. We adopt stochastic gradient descent (SGD) to optimize our model with batch size 100, L2 regularization 10^{-8}, initial learning rate lr 0.2 and the learning rate is decayed[4] with respect to the number of epoch. We select the best model based on the performance on the dev set[5] and apply it to the test set. We use the bootstrapping t-test to compare the results.

Baselines We compare our model with several baselines with or without dependency tree information. The first one is BERT-CRF, where we apply a CRF layer on top of BERT (Devlin et al., 2019). Secondly, we compare with the BERT implementation by HuggingFace (Wolf et al., 2019). For models with dependency trees, we take the models BiLSTM-GCN-CRF and dependency-guided LSTM-CRF (DGLSTM-CRF) proposed by Jie and Lu (2019), and our implemented GCN-BiLSTM-CRF. The BiLSTM-GCN-CRF model simply stacks the GCN on top of the BiLSTM to incorporate the dependency trees. The GCN-BiLSTM-CRF model takes the concatenation of the graph-encoded representation from GCN and word embedding as input into BiLSTM. The DGLSTM-CRF takes the concatenation of the head word representation and word embedding as input into BiLSTM. Note that the original implementation of DGLSTM-CRF uses ELMo (Peters et al., 2018), but we also implement it with BERT. Besides, we compare our model with previous works that have results on these datasets.

[4]We set the decay as 0.1 and the learning rate for each epoch equals to $lr/(1 + decay * (epoch - 1))$.

[5]The experimental results on the dev set and other experimental details can be found in the Appendix.

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 (Ours)	**83.90**	**81.65**	**82.76**	**86.22**	**84.24**	**85.09**
+ Contextualized Word Representation						
BERT-CRF*	76.34	76.05	76.19	79.30	77.22	78.24
Wolf et al. (2019)*	82.82	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 $_{+\text{BERT}}$ (Ours)	**89.07**	**89.04**	**89.05**	**89.66**	**90.54**	**90.10**

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

4.1 Main Results

SemEval 2010 Task 1 Table 2 shows comparisons of our model with baseline models on the SemEval 2010 Task 1 Catalan and Spanish datasets. Our Syn-LSTM-CRF model outperforms all existing models with F_1 82.76 and 85.09 ($p < 10^{-5}$) compared to DGLSTM-CRF on Catalan and Spanish datasets when FastText word embeddings are used. Our model outperforms the BiLSTM-CRF model by 13.25 and 11.22 F_1 points, and outperforms BiLSTM-GCN-CRF (Jie and Lu, 2019) model by 4.64 and 3.16 on Catalan and Spanish. The large performance gap between BiLSTM-GCN-CRF and our model indicates that Syn-LSTM-CRF shows better compatibility with GCN, and this confirms that simply stacking GCN on top of the BiLSTM does not perform well. Our method outperforms GCN-BiLSTM-CRF model

by 5.33 and 3.24 F_1 points on Catalan and Spanish. This shows that our proposed model demonstrates a better integration of contextual information and structured information. Furthermore, our proposed method brings 1.12 and 1.62 F_1 points improvement on Catalan and Spanish datasets compare to the DGLSTM-CRF (Jie and Lu, 2019). The DGLSTM-CRF employs 2-layer dependency guided BiLSTM to capture grandchild dependencies, which leads to longer training time and more model parameters. However, our Syn-LSTM-CRF is able to get better performance with fewer model parameters and shorter training time because of the fewer LSTM layers. Such results demonstrate that our proposed Syn-LSTM-CRF manages to capture structured information effectively.

Furthermore, with the contextualized word representation, the Syn-LSTM-CRF$_{+\text{BERT}}$ achieves much higher performance improvement than any other method. Our model outperforms the strong baseline model DGLSTM-CRF$_{+\text{ELMO}}$ by 4.83 and 2.54 in terms of F_1 ($p < 10^{-5}$) on Catalan and Spanish, respectively.

OntoNotes 5.0 English To understand the generalizability of our model, we evaluate the proposed Syn-LSTM-CRF model on large scale OntoNotes 5.0 datasets. Table 3 shows comparisons of our model with baseline models on English. Our Syn-LSTM-CRF model outperforms all existing methods with 89.04 in terms of F_1 score ($p < 0.01$) compared to DGLSTM-CRF, when GloVE word embeddings are used. Our model outperforms the BiLSTM-CRF model by 1.97 in F_1, BiLSTM-GCN-CRF (Jie and Lu, 2019) model by 0.86. Note that our implemented GCN-BiLSTM-CRF outperforms the previous DGLSTM-CRF (Jie and Lu, 2019) by 0.14 in F_1. Our Syn-LSTM-CRF further brings the improvement to 0.52. Moreover, with the contextualized word representation BERT, our method achieves an F_1 score of 90.85 ($p < 10^{-5}$) compared to DGLSTM-CRF$_{+\text{ELMO}}$. Our method outperforms the previous model (Luo et al., 2020), which relies on document-level information, by 0.55 in F_1. Furthermore, the performance improvement on recall is more prominent as compared to precision. This shows that the proposed Syn-LSTM-CRF is able to extract more entities.

OntoNotes 5.0 Chinese We present the experimental results on the OntoNotes 5.0 Chinese test set in Table 4. Our model still consistently outperforms the baseline models, specifically by 2.04 in F_1 compared to BiLSTM-CRF, by 2.39 compared to BiLSTM-GCN-CRF, by 1.86 compared to GCN-BILSTM-CRF and by 1.11 ($p < 10^{-5}$) compared to DGLSTM-CRF when FastText is used. Note that the baseline BiLSTM-GCN-CRF model is 0.35 points worse than BiLSTM-CRF. Such results further confirm the effectiveness of our proposed Syn-LSTM-CRF for incorporating structured information. We find a similar behavior when the contextualized word representation BERT is used. With the contextualized word representation, we achieve a higher F_1 score of 80.20.

Models	*P.*	*R.*	F_1
Chiu and Nichols (2016a)	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 (Ours)	**88.96**	**89.13**	**89.04**
+ Contextualized Word Representation			
Akbik et al. (2018)	-	-	89.30
BERT-CRF*	88.42	88.33	88.37
Wolf et al. (2019)*	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$_{+\text{BERT}}$ (Ours)	**90.14**	**91.58**	**90.85**

Table 3: Experimental results [%] on OntoNotes 5.0 English test set. The models with * symbol are our implementations. The models with † symbol are retrieved from Jie and Lu (2019). There are also other methods (Li et al., 2020a,b) that use external information, (Yu et al., 2020) use document-level information to encode the sentence, which are not direct comparisons to ours.

5 Analysis

Robustness Analysis To study the robustness of our model and check whether our model can regulate the flow of information from the graph-encoded representation, we analyze the influence of the quality of dependency trees. We train and evaluate an additional dependency parser (Dozat and Manning, 2017). Specifically, we train the

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 (Ours)	77.95	**79.07**	**78.51**
+ Contextualized Word Representation			
BERT-CRF*	**79.83**	79.68	79.75
Wolf et al. (2019)*	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$_{+\text{BERT}}$ (Ours)	78.66	**81.80**	**80.20**

Table 4: Experimental results [%] on OntoNotes 5.0 Chinese test set. The models with * symbol are our implementations. The models with † symbol are retrieved from Jie and Lu (2019). There are also other methods (Li et al., 2020a,b) that use external information, which are not direct comparisons to ours.

dependency parser[6] on the given training datasets and select the best model based on the dev sets. Then we apply the best model to the test sets to obtain dependency trees. We also train and evaluate our model with random dependency trees. Table 8 presents the comparisons between Syn-LSTM-CRF$_{+\text{BERT}}$ and DGLSTM-CRF$_{+\text{ELMO}}$ with given, predicted and random dependency trees. We observe that both models encounter a performance drop when we use the predicted parse trees and random trees. Our performance differences with the given parse trees are relatively smaller than the corresponding differences in DGLSTM-CRF$_{+\text{ELMO}}$. Such an observation demonstrates the robustness of our proposed model against structured information from the trees of different quality. It is worthwhile to note that, with the predicted dependencies, our proposed Syn-LSTM-CRF$_{+\text{BERT}}$ is still able to outperform the strong baseline DGLSTM-CRF$_{+\text{ELMO}}$ even with the given parse trees on Catalan, English, and Chinese datasets.

To further study the robustness, we conduct an analysis to investigate if the gate $\mathbf{m_t}$ (Figure 2) has the ability to regulate the flow of information from the graph-encoded representation. Intuitively, the gate $\mathbf{m}_t$ should tend to have a small value when

[6]The performance of the dependency parser can be found in the Appendix.

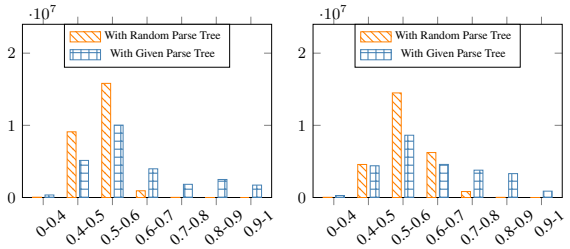

Figure 4: Left: Catalan, Right: Spanish. x-axis: the value of gate $\mathbf{m}_t$. y-axis: the number of words.

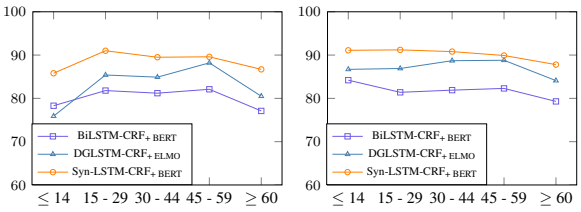

Figure 5: Left: Catalan, Right: Spanish. x-axis: sentence length. y-axis:F_1 score (%). Note that DGLSTM-CRF$_{+\text{ELMO}}$ have better performance compared to DGLSTM-CRF$_{+\text{BERT}}$ based on Table 2, 3, 4.

the quality of the parse tree is not good (e.g., with random trees). We statistically plot the number of words with respect to different gate value ranges ($\mathbf{m}_t$). Figure 4 shows the comparison between the models of using random trees and given trees on Catalan and Spanish[7]. We observe that the gate $\mathbf{m}_t$ is more likely to open (the value is higher) when we use the given parse trees compared with random parse trees. Such behavior demonstrates that our proposed model can selectively aggregate the information from the graph-encoded representation.

Effect of Sentence Length We compare the performance of our Syn-LSTM-CRF$_{+\text{BERT}}$ with BiLSTM-CRF$_{+\text{BERT}}$ and DGLSTM-CRF$_{+\text{ELMO}}$ models with respect to sentence length, and the results are shown in Figure 5. We observe that the Syn-LSTM-CRF$_{+\text{BERT}}$ model consistently outperforms the two baseline models on the four languages[8]. In particular, although the performance tends to drop as the sentence length increases, our proposed model shows relatively better performance when the sentence length is $\geq$ 60. This confirms that the proposed Syn-LSTM-CRF$_{+\text{BERT}}$ is able to effectively incorporate structured information. Note that our 2-layer GCN is computed based on the

[7]We found a similar behavior for OntoNotes 5.0 English and Chinese datasets, and the detailed result can be found in the Appendix.

[8]See the Appendix for the results on OntoNotes 5.0 English and Chinese datasets.

Models	Catalan			Spanish			English			Chinese		
	P.	*R.*	F_1	*P.*	*R.*	F_1	*P.*	*R.*	F_1	*P.*	*R.*	F_1
DGLSTM-CRF$_{+\text{ELMO}}$ (Given)	84.71	83.75	84.22	87.79	87.33	87.56	89.59	90.17	89.88	78.86	81.00	79.92
DGLSTM-CRF$_{+\text{ELMO}}$ (Predicted)	-	-	82.37	-	-	83.92	-	-	89.64	-	-	79.59
Differences	-	-	**-1.85**	-	-	**-3.64**	-	-	**-0.24**	-	-	**-0.33**
DGLSTM-CRF$_{+\text{ELMO}}$ (Random)	78.99	79.31	79.15	82.11	80.89	81.49	88.80	88.91	88.85	77.68	80.60	79.11
Differences	**-5.72**	**-4.44**	**-5.07**	**-5.68**	**-6.44**	**-6.07**	**-0.79**	**-1.26**	**-1.03**	**-1.18**	**-0.40**	**-0.81**
Syn-LSTM-CRF$_{+\text{BERT}}$ (Given)	89.07	89.04	89.05	89.66	90.54	90.10	90.14	91.58	90.85	78.66	81.80	80.20
Syn-LSTM-CRF$_{+\text{BERT}}$ (Predicted)	87.33	87.42	87.38	86.50	87.49	86.99	89.91	91.27	90.58	78.86	81.57	80.19
Differences	**-1.74**	**-1.62**	**-1.67**	**-3.16**	**-3.05**	**-3.11**	**-0.23**	**-0.31**	**-0.27**	**+0.20**	**-0.23**	**-0.01**
Syn-LSTM-CRF$_{+\text{BERT}}$ (Random)	84.57	85.53	85.05	84.61	86.61	85.59	89.24	90.46	89.84	77.25	81.91	79.51
Differences	**-4.50**	**-3.51**	**-4.00**	**-5.05**	**-3.93**	**-4.51**	**-0.90**	**-1.12**	**-1.01**	**-1.41**	**-0.11**	**-0.69**

Table 5: Performance comparison between adopting the given, predicted and random dependencies on SemEval 2010 Task 1 Catalan and Spanish, and OntoNotes 5.0 English and Chinese datasets. Note that DGLSTM-CRF$_{+\text{ELMO}}$ have better performance compared to DGLSTM-CRF$_{+\text{BERT}}$ based on Table 2, 3, 4.

dependency trees, which include both short-range dependencies and long-range dependencies. With the graph-encoded representation and the proposed Syn-LSTM-CRF$_{+\text{BERT}}$, the individual word representation is enhanced by both contextual and structured information. Therefore, for the sentences with length of ≤ 14, we can still observe obvious improvements. The significant performance improvements on the four datasets show the capability of our Syn-LSTM-CRF to capture the structured information despite the sentence length.

Effect of Entity Length We conduct another evaluation on BiLSTM-CRF$_{+\text{BERT}}$, DGLSTM-CRF$_{+\text{ELMO}}$, and Syn-LSTM-CRF$_{+\text{BERT}}$ models with respect to entity length $\in \{1, 2, 3, 4, 5, \geq 6\}$ on the four languages. Table 6 shows the performance comparison of two models with respect to entity length. With the structured information, both DGLSTM-CRF$_{+\text{ELMO}}$ and Syn-LSTM-CRF$_{+\text{BERT}}$ achieve better performance compared to BiLSTM-CRF$_{+\text{BERT}}$. When the length of entity is ≤ 3, Syn-LSTM-CRF$_{+\text{BERT}}$ achieves better results compared to DGLSTM-CRF$_{+\text{ELMO}}$. This confirms that our proposed method can effectively incorporate the structured information. Our model consistently outperforms BiLSTM-CRF$_{+\text{BERT}}$, and the performance tends to have more improvements when entities are getting longer except on the Chinese dataset. We note there are some special characteristics of the Chinese language. As mentioned by Jie and Lu (2019), the percentage of entities that are able to perfectly form a sub-tree is only 92.9% for OntoNotes Chinese, as compared to 98.5%, 100%, 100% for OntoNotes English, SemEval Catalan and Spanish. Furthermore, the ratio of long entities is much higher for Catalan and Spanish compared to English and Chinese. The experimental results on Catalan and Spanish datasets show significant improvements for long entities. Such results show that the structured information conveyed by the dependency trees can be more crucial when entity length becomes longer.

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**
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
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**
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

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 in the main paper.

Number of GCN Layers To fully explore the impact of the number of GCN layers, we conduct another experiment on Syn-LSTM-CRF$_{+\text{BERT}}$ model with the number of GCN layers $\in \{1, 2, 3\}$, and Figure 6 shows the performance on the dev set of the four languages. The last bar, indicated as AVG, is obtained by averaging the dev results on the four datasets. We observe that the overall performance is better when the number of GCN layers equals 2. Note that similar behavior can also be found in the work by Kipf and Welling (2017) for document classification and node classification. Therefore, we evaluate our proposed Syn-LSTM-CRF model with 2-layer GCN.

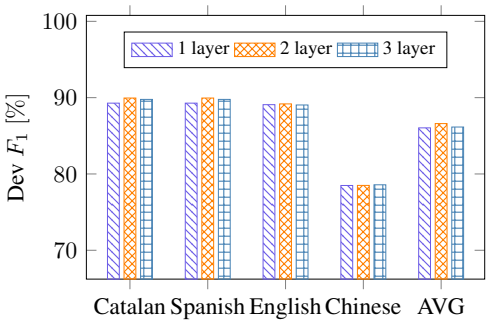

Figure 6: Performance of different number of layers of GCN on dev set.

Model	$P.$	$R.$	F_1
Syn-LSTM-CRF$_{+\text{BERT}}$	**90.14**	**91.58**	**90.85**
– 1 layer GCN	89.93	91.30	90.61
– 2 layer GCN	89.50	89.93	89.72
– original dependency	89.91	91.27	90.58
– dependency embedding	89.85	91.31	90.58
– POS embedding	89.84	90.95	90.46

Table 7: Ablation study of the Syn-LSTM-CRF$_{+\text{BERT}}$ model on OntoNotes 5.0 English. – means removing.

Ablation Study To understand the contribution of each component, we conduct an ablation study on the OntoNotes 5.0 English dataset, and Table 7 presents the detailed results of our model with contextualized representation. We find that the performance drops by 0.24 F_1 score when we only use 1-layer GCN. Without GCN at all, the score drops by 1.13 F_1. The original dependency contributes 0.27 F_1 score. Removing the dependency relation embedding also decreases the performance by 0.27 F_1. When we remove the POS tags embedding, the result drops by 0.39 F_1.

6 Related Work

LSTM LSTM has demonstrated its great effectiveness in many NLP tasks and becomes a standard module for many state-of-the-art models (Wen et al., 2015; Ma and Hovy, 2016; Dozat and Manning, 2017). However, the sequential nature of the LSTM makes it challenging to capture long-range dependencies. Zhang et al. (2018a) propose the S-LSTM model to include a sentence state to allow both local and global information exchange simultaneously. Mogrifier LSTM (Melis et al., 2020) mutually gates the current input and the previous output to enhance the interaction between the input and the context. These two works do not consider structured information for the LSTM design. Since natural language is usually structured, Shen et al. (2018) propose ON-LSTM to add a hierarchical bias to allow the neurons to be updated by following certain order. While the ON-LSTM is learning the latent constituency parse trees, we focus on incorporating the explicit structured information conveyed by the dependency parse trees.

NER Early work (Sasano and Kurohashi, 2008) uses syntactic dependency features to improve the SVM performance on Japanese NER task. Liu et al. (2010) propose to construct skip-edges to link similar words or words having typed dependencies to capture long-range dependencies. The later works (Collobert et al., 2010; Lample et al., 2016; Chiu and Nichols, 2016b) focus on using neural networks to extract features and achieved the state-of-the-art performance. Jie et al. (2017) find that some relations between the dependency edges and the entities can be used to reduce the search space of their model, which significantly reduces the time complexity. Yu et al. (2020) employ pre-trained language model to encode document-level information to explore all spans with the graph-based dependency graph based ideas. The pre-trained language models (e.g., BERT (Devlin et al., 2019), ELMO (Peters et al., 2018)) further improve neural-based approaches with a good contextualized representation. However, previous works did not focus on investigating how to effectively integrate structured and contextual information well.

7 Conclusion

In this paper, we propose a simple and robust Syn-LSTM model to better integrate the structured information leveraged from the long-range dependencies. Specifically, we introduce an additional graph-encoded representation to each recurrent unit. Such a graph-encoded representation can be obtained via GNNs. Through the newly designed gating mechanism, the hidden states are enhanced by contextual information captured by the linear sequence and structured information captured by the dependency trees. We present the Syn-LSTM-CRF for NER and adopt the GCN on dependency trees to obtain the graph-encoded representations. Our extensive experiments and analysis on the datasets with four languages demonstrate that the proposed Syn-LSTM is able to effectively incorporate both contextual and structured information. The robustness analysis demonstrates that our model is capable of selectively aggregating the information from the graph-encoded representation.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. This research is partially supported by Ministry of Education, Singapore, under its Academic Research Fund (AcRF) Tier 2 Programme (MOE AcRF Tier 2 Award No: MOE2017-T2-1-156). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of the Ministry of Education, Singapore.

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 labeling. In *Proceedings of COLING*.

Y. Bengio. 2009. *Learning Deep Architectures for AI*. Now Publishers.

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

Jason P.C. Chiu and Eric Nichols. 2016b. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*.

N. Chomsky. 1956. Three models for the description of language. *IRE Transactions on Information Theory*.

N. Chomsky. 1969. *Aspects of the Theory of Syntax*. MIT Press.

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2010. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*.

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*.

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*.

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*.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017a. Representation learning on graphs: Methods and applications. *IEEE Data Eng. Bull.*

William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017b. Inductive representation learning on large graphs. In *Proceedings of NIPS*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*.

Zhanming Jie and Wei Lu. 2019. Dependency-guided lstm-crf for named entity recognition. In *Proceedings of EMNLP*.

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*.

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*.

Omer Levy, Kenton Lee, Nicholas FitzGerald, and Luke Zettlemoyer. 2018. Long short-term memory as a dynamically computed element-wise weighted sum. In *Proceedings of ACL*.

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*.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020a. A unified mrc framework for named entity recognition. In *Proceedings of ACL*.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020b. Dice loss for data-imbalanced nlp tasks. In *Proceedings of ACL*.

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*.

Ying Luo, Fengshun Xiao, and Hai Zhao. 2020. Hierarchical contextualized representation for named entity recognition. In *Proceedings of AAAI*.

Xuezhe Ma and Eduard H. 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*.

Gábor Melis, Tomás Kociský, and Phil Blunsom. 2020. Mogrifier lstm. In *Proceedings of ICLR*.

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. 2018. Deep contextualized word representations. In *Proceedings of NAACL*.

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*.

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*.

D. Sandra and M. Taft. 2014. *Morphological Structure, Lexical Representation and Lexical Access (RLE Linguistics C: Applied Linguistics): A Special Issue of Language and Cognitive Processes*. Taylor & Francis.

Ryohei Sasano and Sadao Kurohashi. 2008. Japanese named entity recognition using structural natural language processing. In *Proceedings of IJCNLP*.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron C. Courville. 2018. Ordered neurons: Integrating tree structures into recurrent neural networks. In *Proceedings of ICLR*.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of EMNLP*.

Bailin Wang and Wei Lu. 2018. Neural segmental hypergraphs for overlapping mention recognition. In *Proceedings of EMNLP*.

Bailin Wang, Wei Lu, Yu Wang, and Hongxia Jin. 2018. A neural transition-based model for nested mention recognition. In *Proceedings of EMNLP*.

Rui Wang, Xin Xin, Wei Chang, Kun Ming, Biao Li, and Xin Fan. 2019. Chinese ner with height-limited constituent parsing. In *Proceedings of AAAI*.

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*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei hao Su, David Vandyke, and Steve J. Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of EMNLP*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*.

Han Xiao. 2018. bert-as-service. `https://github.com/hanxiao/bert-as-service`.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of ACL*.

Yue Zhang, Qi Liu, and Linfeng Song. 2018a. Sentence-state lstm for text representation. In *Proceedings of ACL*.

Yue Zhang and Jie Yang. 2018. Chinese NER using lattice LSTM. In *Proceedings of ACL*.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018b. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of EMNLP*.

A Experimental details

We test our model on RTX 2080 Ti GPU and Nvidia Tesla V100 GPU, with CUDA version 10.1, PyTorch version 1.40. The average run time for Syn-LSTM is 52 sec/epoch, 55 sec/epoch, 290 sec/epoch, 350 sec/epoch for Catalan, Spanish, Chinese and English datasets respectively. The total number of parameters is 11M. Table 10 shows the performance of our model on the dev sets of OntoNotes 5.0 English and Chinese, SemEval 2010 Task 1 Catalan and Spanish.

For hyper-parameter, we use the FastText (Grave et al., 2018) 300 dimensional embeddings to initialize the word embeddings for Catalan, Spanish, and Chinese. 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

	English	Chinese	Catalan	Spanish
Dependency LAS†	94.89	89.28	93.25	93.35

Table 8: Performance of the trained dependency parser.

Dataset		Entity Length					
		1	2	3	4	5	≥6
English	Train	46,525	17,391	9,714	4,892	1,938	1,368
	Dev	6,325	2,395	1,256	643	275	172
	Test	6,129	2,598	1,359	706	278	187
Chinese	Train	47,285	9,668	3,626	1,139	467	358
	Dev	6,969	1,397	473	169	55	41
	Test	5,479	1299	473	146	55	42
Catalan	Train	8,819	3,897	1,742	264	119	437
	Dev	1,370	676	269	40	18	58
	Test	1,601	811	338	57	27	76
Spanish	Train	10,307	3,609	2,302	301	175	603
	Dev	1,523	559	348	54	31	100
	Test	1,755	702	369	59	34	127

Table 9: Number of entities with respect to entity length for OntoNotes 5.0 English and Chinese, SemEval 2010 Catalan and Spanish datasets.

language model BERT (Devlin et al., 2019) for the four datasets. Specifically, we use bert-as-service (Xiao, 2018) to generate the contextualized word representation without fine-tuning. Following Luo et al. (2020), we select the 18^{th} layer of the cased version of BERT large model for the experiments on the OntoNotes 5.0 English data. We use the the 9^{th} layer of cased version of BERT base model for the experiments on the rest three datasets. For the character embedding, we randomly initialize the character embeddings and set the dimension as 30, and set the hidden size of character-level BiLSTM as 50. The hidden size of GCN and Syn-LSTM is set as 200. Note that we only use one layer of bi-directional Syn-LSTM for our experiments. Dropout is set to 0.5 for input embeddings and hidden states. We adopt stochastic gradient descent (SGD) to optimize our model with batch size 100, L2 regularization 10^{-8}, learning rate 0.2 and the learning rate is decayed with respect to the number of epoch [9] .

B Performance of dependency parser

Table 8 presents the performance of dependency parser.

[9]We set the decay as 0.1 and the learning rate for each epoch equals to $learning\_rate/(1 + decay * (epoch - 1))$.

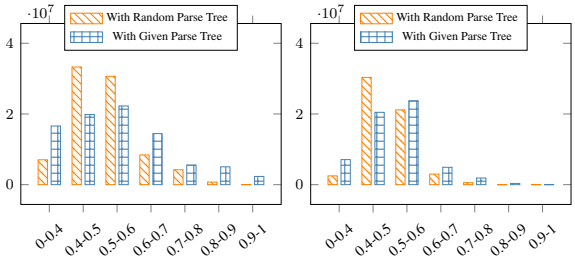

Figure 7: Left: English, Right: Chinese. The x-axis indicates the value of gate m_t, the y-axis denotes the number of cells.

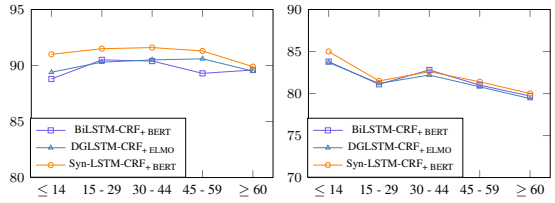

Figure 8: Left: English, Right: Chinese. x-axis: sentence length. y-axis:F_1 score (%). Note that DGLSTM-CRF$_{+\text{ELMO}}$ have better performance compared to DGLSTM-CRF$_{+\text{BERT}}$ based on the results in the main paper.

C More data statistics

Table 9 shows the statistics of the number of entities with respect to entity length for OntoNotes 5.0 English and Chinese, SemEval 2010 Task 1 Catalan and Spanish datasets.

D More Robustness Analysis

Figure 7 shows the comparisons of the models of using random trees and given trees on OntoNotes 5.0 English and Chinese datasets.

E Effect of Sentence Length

We compare the performance of our Syn-LSTM-CRF$_{+\text{BERT}}$ with BiLSTM-CRF$_{+\text{BERT}}$ and DGLSTM-CRF$_{+\text{ELMO}}$ models with respect to sentence length, and the results are shown in Figure 8.

F Case Study

We further show an example to visualize the propagation of non-local information (Figure 9). The example is selected from OntoNotes 5.0 English dataset. Even though the DGLSTM-CRF (Jie and Lu, 2019) model is able to recognize "Tianshui" as a named entity, it predicts a wrong entity type as PERSON while the true type is GPE. If only looking at the first half of the sentence, it is possible to predict "Tianshui" as PERSON because of the local information "age". However, the second half of the sentence confirms that the entity type of

Models	English			Chinese			Catalan			Spanish		
	P.	*R.*	F_1	*P.*	*R.*	F_1	*P.*	*R.*	F_1	*P.*	*.R*	F_1
Syn-LSTM-CRF	86.73	87.71	87.22	77.25	75.74	76.49	84.48	82.60	83.53	83.76	82.22	82.98
Syn-LSTM-CRF$_{+\text{BERT}}$	88.10	90.27	89.17	78.05	78.84	78.45	89.87	89.76	89.81	88.50	88.60	88.55

Table 10: Experimental results [%] on dev set.

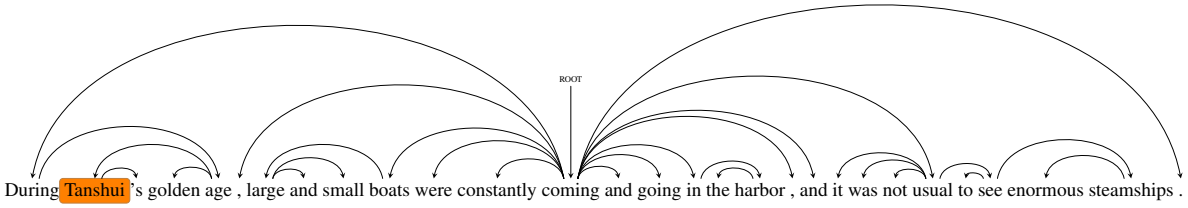

Figure 9: An example of dependency tree. The mentioned entity is highlighted in orange, and the entity type is GPE.

"Tianshui" is GPE. With the non-local information from the graph-encoded representation, our Syn-LSTM-CRF successfully predicts the right entity type.