Hierarchically-Refined Label Attention Network for Sequence Labeling

Leyang Cui[1,2,3] and **Yue Zhang**[2,3]

[1]Zhejiang University

[2]School of Engineering, Westlake University

[3]Institute of Advanced Technology, Westlake Institute for Advanced Study

`cuileyang@westlake.edu.cn, yue.zhang@wias.org.cn`

Abstract

CRF has been used as a powerful model for statistical sequence labeling. For neural sequence labeling, however, BiLSTM-CRF does not always lead to better results compared with BiLSTM-softmax local classification. This can be because the simple Markov label transition model of CRF does not give much information gain over strong neural encoding. For better representing label sequences, we investigate a hierarchically-refined label attention network, which explicitly leverages label embeddings and captures potential long-term label dependency by giving each word incrementally refined label distributions with hierarchical attention. Results on POS tagging, NER and CCG supertagging show that the proposed model not only improves the overall tagging accuracy with similar number of parameters, but also significantly speeds up the training and testing compared to BiLSTM-CRF.

1 Introduction

Conditional random fields (CRF) (Lafferty et al., 2001) is a state-of-the-art model for statistical sequence labeling (Toutanova et al., 2003; Peng et al., 2004; Ratinov and Roth, 2009). Recently, CRF has been integrated with neural encoders as an output layer to capture label transition patterns (Zhou and Xu, 2015; Ma and Hovy, 2016). This, however, sees mixed results. For example, previous work (Reimers and Gurevych, 2017; Yang et al., 2018) has shown that BiLSTM-softmax gives better accuracies compared to BiLSTM-CRF for part-of-speech (POS) tagging. In addition, the state-of-the-art neural Combinatory Categorial Grammar (CCG) supertaggers do not use CRF (Xu et al., 2015; Lewis et al., 2016).

One possible reason is that the strong representation power of neural sentence encoders such as BiLSTMs allow models to capture implicit long-range label dependencies from input
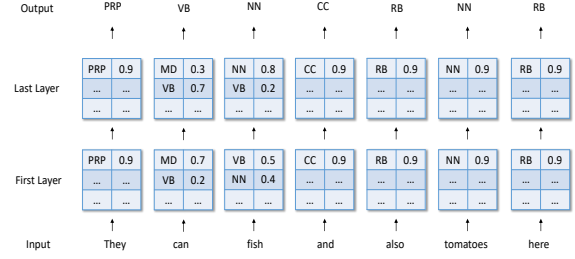

Figure 1: Visualization of hierarchically-refined Label Attention Network for POS tagging. The numbers indicate the label probability distribution for each word.

word sequences alone (Kiperwasser and Goldberg, 2016; Dozat and Manning, 2016; Teng and Zhang, 2018), thereby allowing the output layer to make local predictions. In contrast, though explicitly capturing output label dependencies, CRF can be limited by its Markov assumptions, particularly when being used on top of neural encoders. In addition, CRF can be computationally expensive when the number of labels is large, due to the use of Viterbi decoding.

One interesting research question is whether there is neural alternative to CRF for neural sequence labeling, which is both faster and more accurate. To this question, we investigate a neural network model for output label sequences. In particular, we represent each possible label using an embedding vector, and aim to encode sequences of label distributions using a recurrent neural network. One main challenge, however, is that the number of possible label sequences is exponential to the size of input. This makes our task essentially to represent a full-exponential search space without making Markov assumptions.

We tackle this challenge using a hierarchically-refined representation of marginal label distributions. As shown in Figure 1, our model consists of a multi-layer neural network. In each layer, each input words is represented together with its

marginal label probabilities, and a sequence neural network is employed to model unbounded dependencies. The marginal distributions space are refined hierarchically bottom-up, where a higher layer learns a more informed label sequence distribution based on information from a lower layer.

For instance, given a sentence "They$_1$ can$_2$ fish$_3$ and$_4$ also$_5$ tomatoes$_6$ here$_7$", the label distributions of the words *can$_2$* and *fish$_3$* in the first layer of Figure 1 have higher probabilities on the tags MD (modal verb) and VB (base form verb), respectively, though not fully confidently. The initial label distributions are then fed as the inputs to the next layer, so that long-range label dependencies can be considered. In the second layer, the network can learn to assign a noun tag to *fish$_3$* by taking into account the highly confident tagging information of *tomatoes$_6$ (NN)*, resulting in the pattern "*can$_2$ (VB) fish$_3$ (NN)*".

As shown in Figure 2, our model consists of stacked attentive BiLSTM layers, each of which takes a sequence of vectors as input and yields a sequence of hidden state vectors together with a sequence of label distributions. The model performs attention over label embeddings (Wang et al., 2015; Zhang et al., 2018a) for deriving a marginal label distributions, which are in turn used to calculate a weighted sum of label embeddings. Finally, the resulting packed label vector is used together with input word vectors as the hidden state vector for the current layer. Thus our model is named *label attention network* (LAN). For sequence labeling, the input to the whole model is a sentence and the output is the label distributions of each word in the final layer.

BiLSTM-LAN can be viewed as a form of multi-layered BiLSTM-softmax sequence labeler. In particular, a single-layer BiLSTM-LAN is identical to a single-layer BiLSTM-softmax model, where the label embedding table serves as the softmax weights in BiLSTM-softmax, and the label attention distribution is the softmax distribution in BiLSTM-softmax. The traditional way of making a multi-layer extention to BiLSTM-softmax is to stack multiple BiLSTM encoder layers before the softmax output layer, which learns a deeper input representation. In contrast, a multi-layer BiLSTM-LAN stacks both the BiLSTM encoder layer and the softmax output layer, learning a deeper representation of both the input and candidate output sequences.

On standard benchmarks for POS tagging, NER and CCG supertagging, our model achieves significantly better accuracies and higher efficiencies than BiLSTM-CRF and BiLSTM-softmax with similar number of parameters. It gives highly competitive results compared with top-performance systems on WSJ, OntoNotes 5.0 and CCGBank without external training. In addition to accuracy and efficiency, BiLSTM-LAN is also more interpretable than BiLSTM-CRF thanks to visualizable label embeddings and label distributions. Our code and models are released at `https://github.com/Nealcly/LAN`.

2 Related Work

Neural Attention. Attention has been shown useful in neural machine translation (Bahdanau et al., 2014), sentiment classification (Chen et al., 2017; Liu and Zhang, 2017), relation classification (Zhou et al., 2016), read comprehension (Hermann et al., 2015), sentence summarization (Rush et al., 2015), parsing (Li et al., 2016), question answering (Wang et al., 2018b) and text understanding (Kadlec et al., 2016). Self-attention network (SAN) (Vaswani et al., 2017) has been used for semantic role labeling (Strubell et al., 2018), text classification (Xu et al., 2018; Wu et al., 2018) and other tasks. Our work is similar to Vaswani et al. (2017) in the sense that we also build a hierarchical attentive neural network for sequence representation. The difference lies in that our main goal is to investigate the encoding of exponential *label* sequences, whereas their work focuses on encoding of a *word* sequence only.

Label Embeddings. Label embedding was first used in the field of computer vision for facilitating zero-shot learning (Palatucci et al., 2009; Socher et al., 2013; Zhang and Saligrama, 2016). The basic idea is to improve the performance of classifying previously unseen class instances by learning output label knowledge. In NLP, label embeddings have been exploited for better text classification (Tang et al., 2015; Nam et al., 2016; Wang et al., 2018a). However, relatively little work has been done investigating label embeddings for sequence labeling. One exception is Vaswani et al. (2016b), who use supertag embeddings in the output layer of a CCG supertagger, through a combination of local classification model rescored using a supertag language model. In contrast, we model deep label interactions using a dynamically refined
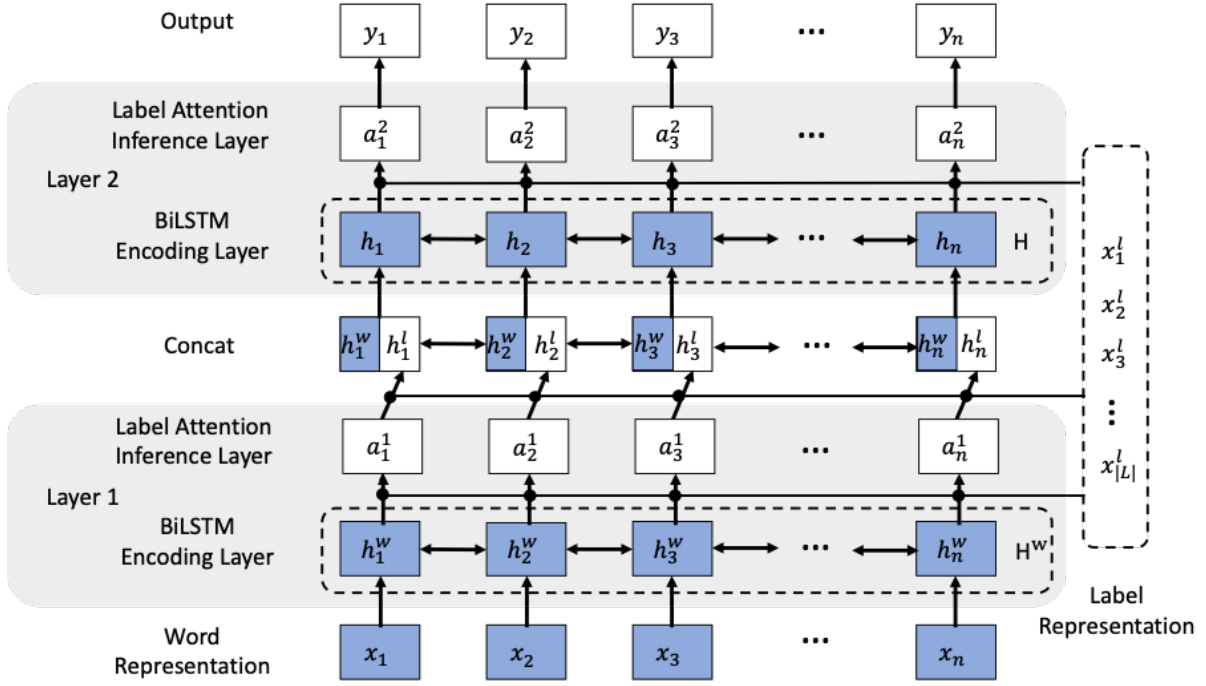
Figure 2: Architecture of hierarchically-refined label attention network.

sequence representation network. To our knowledge, we are the first to investigate a hierarchical attention network over a label space.

Neural CRF. There has been methods that aim to speed up neural CRF (Tu and Gimpel, 2018), and to solve the Markov constraint of neural CRF. In particular, Zhang et al. (2018b) predicts a sequence of labels as a sequence to sequence problem; Guo et al. (2019) further integrates global input information in encoding. Capturing non-local dependencies between labels, these methods, however, are slower compared with CRF. In contrast to these lines of work, our method is both asymptotically faster and empirically more accurate compared with neural CRF.

3 Baseline

We implement BiLSTM-CRF and BiLSTM-softmax baseline models with character-level features (Dos Santos and Zadrozny, 2014; Lample et al., 2016), which consist of a word representation layer, a sequence representation layer and a local softmax or CRF inference layer.

3.1 Word Representation Layer

Following Dos Santos and Zadrozny (2014) and Lample et al. (2016), we use character information for POS tagging. Given a word sequence

$w_1, w_2, ..., w_n$, where c_{ij} denotes the jth character in the ith word, each c_{ij} is represented using

$$\mathbf{x}_{ij}^c = \mathbf{e}^c(c_{ij})$$

where $\mathbf{e}^c$ denotes a character embedding lookup table.

We adopt BiLSTM for character encoding. x_i^c denotes the output of character-level encoding.

A word is represented by concatenating its word embedding and its character representation:

$$\mathbf{x}_i = [\mathbf{e}^w(w_i); \mathbf{x}_i^c]$$

where $\mathbf{e}^w$ denotes a word embedding lookup table.

3.2 Sequence Representation Layer

For sequence encoding, the input is a sentence $\mathbf{x} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$. Word representations are fed into a BiLSTM layer, yielding a sequence of forward hidden states $\{\overrightarrow{\mathbf{h}_1^w}, \cdots, \overrightarrow{\mathbf{h}_n^w}\}$ and a sequence of backward hidden states $\{\overleftarrow{\mathbf{h}_1^w}, \cdots, \overleftarrow{\mathbf{h}_n^w}\}$, respectively. Finally, the two hidden states are concatenated for a final representation

$$\mathbf{h}_i^w = [\overrightarrow{\mathbf{h}_i^w}; \overleftarrow{\mathbf{h}_i^w}]$$
$$\mathbf{H}^w = \{\mathbf{h}_1^w, \cdots, \mathbf{h}_n^w\}$$

3.3 Inference Layer

CRF. A CRF layer is used on top of the hidden vectors $\mathbf{H}^w$. The conditional probabilities of label distribution sequences $y = \{y_1, \cdots, y_n\}$ is

$$P(y|x) = \frac{\exp(\sum_i(\mathbf{W}_{CRF}^{l_i}\mathbf{h_i}^w + b_{CRF}^{(l_{i-1},l_i)}))}{\sum_{y'}\exp(\sum_i(\mathbf{W}_{CRF}^{l_i'}\mathbf{h_i}^w + b_{CRF}^{(l_{i-1}',l_i')}))}$$

Here y' represents an arbitrary label distribution sequence, $\mathbf{W}_{CRF}^{l_i}$ is a model parameter specific to l_i, and $b_{CRF}^{(l_{i-1}',l_i')}$ is a bias specific to l_{i-1} and l_i.

The first-order Viterbi algorithm is used to find the highest scored label sequence over an input word sequence during decoding.

Softmax. Independent local softmax classification can give competitive result on sequence labeling (Reimers and Gurevych, 2017). For each node, $\mathbf{h}_i^w$ is fed to a softmax layer to find

$$\hat{y}_i = \text{softmax}(\mathbf{W}\mathbf{h}_i^w + \mathbf{b}) \tag{1}$$

where $\hat{y}_i$ is the predicted label for w_i; $\mathbf{W}$ and $\mathbf{b}$ are the parameters for softmax layer.

4 Label Attention Network

The structure of our refined label attention network is shown in Figure 2. We denote our model as BiLSTM-LAN (namely BiLSTM-label attention network) for the remaining of this paper. We use the same input word representations for BiLSTM-softmax, BiLSTM-CRF and BiLSTM-LAN, as described in Section 3.1. Compared with the baseline models, BiLSTM-LAN uses a set of BiLSTM-LAN layers for both encoding and label prediction, instead of a set of traditional sequence encoding layers and an inference layer.

4.1 Label Representation.

Given the set of candidates output labels $L = \{l_1, \cdots, l_{|L|}\}$, each label l_k is represented using an embedding vector:

$$\mathbf{x}_k^l = \mathbf{e}^l(l_k)$$

where $\mathbf{e}^l$ denotes a label embedding lookup table. Label embeddings are randomly initialized and tuned during model training.

4.2 BiLSTM-LAN Layer

The model in Figure 2 consists of 2 BiLSTM-LAN layers. As discussed earlier, each BiLSTM-LAN layer is composed of a BiLSTM encoding sublayer and a label-attention inference sublayer. In paticular, the former is the same as the BiLSTM layer in the baseline model, while the latter uses multi-head attention (Vaswani et al., 2017) to jointly encode information from the word representation subspace and the label representation subspace. For each head, we apply a dot-product attention

with a scaling factor to the inference component, deriving label distributions for BiLSTM hidden states and label embeddings.

BiLSTM Encoding Sublayer. Denote the input to each layer as $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$. BiLSTM (Section 3.2) is used to calculate $\mathbf{H}^w \in \mathbb{R}^{n \times d_h}$, where n and d_h denote the word sequence length and BiLSTM hidden size (same as the dimension of label embedding), respectively.

Label-Attention Inference Sublayer. For the label-attention inference sublayer, the attention mechanism produces an attention matrix $\boldsymbol{\alpha}$ consisting of a potential label distribution for each word. We define $\mathbf{Q} = \mathbf{H}^w$, $\mathbf{K} = \mathbf{V} = \mathbf{x}^l$. $\mathbf{x}^l \in \mathbb{R}^{|L| \times d_h}$ is the label set representation, where $|L|$ is the total number of labels. As shown in Figure 2, outputs are calculated by

$$\mathbf{H}^l = \text{attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \boldsymbol{\alpha}\mathbf{V}$$
$$\boldsymbol{\alpha} = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_h}})$$

Instead of the standard attention mechanism above, it can be beneficial to use multi-head for capturing multiple possible of potential label distributions in parallel.

$$\mathbf{H}^l = \text{concat}(\text{head}, \ldots, \text{head}_k) + \mathbf{H}^w$$
$$\text{head}_i = \text{attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)$$

where $\mathbf{W}_i^Q \in \mathbb{R}^{d_h \times \frac{d_h}{k}}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_h \times \frac{d_h}{k}}$ and $\mathbf{W}_i^V \in \mathbb{R}^{d_h \times \frac{d_h}{k}}$ are parameters to be learned during the training, k is the number of parallel heads.

The final representation for each BiLSTM-LAN layer is the concatenation of BiLSTM hidden states and attention outputs :

$$\mathbf{H} = [\mathbf{H}^w; \mathbf{H}^l]$$

$\mathbf{H}$ is then fed to a subsequent BiLSTM-LAN layer as input, if any.

Output. In the last layer, BiLSTM-LAN directly predicts the label of each word based on the attention weights.

$$\begin{bmatrix} \hat{y_1}^1 \cdots \hat{y_1}^{|L|} \\ \cdots\cdots \\ \cdots\cdots \\ \cdots\cdots \\ \hat{y_i}^1 \cdots \hat{y_i}^{|L|} \end{bmatrix} = \boldsymbol{\alpha}$$

$$\hat{y_i} = \text{argmax}_j(\hat{y_i}^1, ..., \hat{y_i}^n)$$

where $\hat{y}_i{}^j$ denotes the jth candidate label for the ith word and $\hat{y}_i$ is the predicted label for ith word in the sequence.

4.3 Training

BiLSTM-LAN can be trained by standard back-propagation using the log-likelihood loss. The training object is to minimize the cross-entropy between y_i and $\hat{y}_i$ for all labeled gold-standard sentences. For each sentence,

$$L = -\sum_i \sum_j y_i^j \log \hat{y}_i^j$$

where i is the word index, j is the index of labels for each word.

4.4 Complexity

For decoding, the asymptotic time complexity is $O(|L|^2 n)$ and $O(|L|n)$ for BiLSTM-CRF and BiLSTM-LAN, respectively, where $|L|$ is the total number of labels and n is the sequence length. Compared with BiLSTM-CRF, BiLSTM-LAN can significantly increase the speed for sequence labeling, especially for CCG supertagging, where the sequence length can be much smaller than the total number of labels.

4.5 BiLSTM-LAN and BiLSTM-softmax

As mentioned in the introduction, a single-layer BiLSTM-LAN is identical to a single-layer BiLSTM-softmax sequence labeling model. In particular, the BiLSTM-softmax model is given by Eq 1. In a BiLSTM-LAN model, we arrange the set of label embeddings into a matrix as follows:

$$\mathbf{x}^l = [x_1^l; x_2^l; ...; x_{|L|}^l]$$

A naive attention model over X has:

$$\boldsymbol{\alpha} = \text{softmax}(\mathbf{H}\mathbf{x}^l)$$

It is thus straightforward to see that the label embedding matrix $\mathbf{x}^l$ corresponds to the weight matrix $\mathbf{W}$ is Eq 1, and the distribution $\boldsymbol{\alpha}$ corresponds to y in Eq 1.

5 Experiments

We empirically compare BiLSTM-CRF, BiLSTM-softmax and BiLSTM-LAN using different sequence labelling tasks, including English POS tagging, multilingual POS tagging, NER and CCG supertagging.

Data		training	dev	test
WSJ	#l	45	45	45
	#s	38,219	5,527	5,462
	#t	912,344	131,768	129,654
UD_en	#l	50	50	49
	#s	12,544	2,003	2,078
	#t	204,607	25,150	25,097
OntoNotes	#l	18	18	18
	#s	59,924	8,528	8,262
	#t	1,088,503	147,724	152,728
CCGBank	#l	426	323	348
	#s	39,604	1,913	2,407
	#t	929,552	45,422	55,371

Table 1: Data statistics. l:label, s:sentence, t:tokens.

Model	# E /# H	# L	Acc	# Param
	200	1	97.56	5.1M
	400	1	97.57	5.5M
BiLSTM-CRF	400	2	97.57	6.4M
	400	3	97.52	7.4M
	600	2	97.57	8.2M
	600	3	97.50	10.4M
	200	3	97.53	5.7M
	400	2	97.57	8.1M
BiLSTM-LAN	400	3	97.63	10.0M
	400	4	97.60	12.2M
	600	3	97.62	16.5M

Table 2: WSJ development set. E: label embedding size, H: hidden size, L: number of layers.

Model	Accuracy (%)
BiLSTM-CRF	97.57
BiLSTM-softmax	97.58
BiLSTM-LAN w/o attention[†]	97.59
BiLSTM-LAN	97.65

Table 3: Effect of attention layer. † denotes the model without attention sublayers except for the last layer.

5.1 Dataset

For English POS tagging, we use the Wall Street Journal (WSJ) portion of the Penn Treebank (PTB) (Marcus et al., 1993), which has 45 POS tags. We adopt the same partitions used by previous work (Manning, 2011; Yang et al., 2018), selecting sections 0-18 as the training set, sections 19-21 as the development set and sections 22-24 as the test set. For multilingual POS tagging, we use treebanks from Universal Dependencies(UD) v2.2 (Silveira et al., 2014; Nivre et al., 2018) with the standard splits, choosing 8 resource-rich languages in our experiments. For NER, we use the OntoNotes 5.0 (Hovy et al., 2006; Pradhan et al., 2013). Following previous work, we adopt the official OntoNotes data split used for co-reference resolution in the CoNLL-2012 shared task (Pradhan et al., 2012). We train our CCG supertagging

Model	train (s)	test (st/s)
BiLSTM-CRF (POS)	181.32	781.90
BiLSTM-LAN (POS)	128.75	805.32
BiLSTM-CRF (CCG)	884.67	599.18
BiLSTM-LAN (CCG)	369.98	713.70

Table 4: Comparison of the training time for one iteration and decoding speed. st indicates sentences

model on CCGBank (Hockenmaier and Steedman, 2007). Sections 2-21 are used for training, section 0 for development and section 23 as the test set. The performance is calculated on the 425 most frequent labels. Table 1 shows the numbers of sentences, tokens and labels for training, development and test, respectively.

5.2 Settings

Hyper-Parameters. We use 100-dimensional GloVe (Pennington et al., 2014) word embeddings for English POS tagging (WSJ, UD v2.2 EN) and name entity recognition, 300-dimensional Glove word embeddings for CCG supertagging and 64-dimensional Polyglot (Al-Rfou et al., 2013) word embeddings for multilingual POS tagging. The detail values of the hyper-parameters for all experiments are summarized in Appendix A.

Evaluation. F-1 score are used for NER. Other tasks are evaluated based on the accuracy. We repeat the same experiment five times with the same hyperparameters and report the max accuracy, average accuracy and standard deviation for POS tagging. For fair comparison, all experiments are implemented in NCRF++ (Yang and Zhang, 2018) and conducted using a GeForce GTX 1080Ti with 11GB memory.

5.3 Development Experiments

We report a set of WSJ development experiments to show our investigation on key configurations of BiLSTM-LAN and BiLSTM-CRF.

Label embedding size. Table 2 shows the effect of the label embedding size. Notable improvement can be achieved when the label embedding size increases from 200 to 400 in our model, but the accuracy does not further increase when the size increases beyond 400. We fix the label embedding size to 400 for our model.

Number of Layers. For BiLSTM-CRF, previous work has shown that one BiLSTM layer is the most effecitve for POS tagging (Ma and Hovy, 2016; Yang et al., 2018). Table 2 compares different numbers of BiLSTM layers and hidden sizes.

Model	Mean±std	Max
BiLSTM-CRF[†]	97.47±0.02	97.49
BiLSTM-CRF[‡]	97.50±0.03	97.51
BiLSTM-softmax[†]	97.48±0.02	97.51
BiLSTM-LAN	**97.58±0.04**	**97.65**

Table 5: Result for POS tagging on WSJ. † Yang et al. (2018) and ‡ Yasunaga et al. (2018) are baseline models re-implemented in NCRF++ (Yang and Zhang, 2018). Our results are same as Table 6 of Yang et al. (2018).

Model	Accuracy
Plank et al. (2016)	97.22
Huang et al. (2015)	97.55
Ma and Hovy (2016)	97.55
Liu et al. (2017)	97.53
Yang et al. (2018)	97.51
Zhang et al. (2018c)	97.55
Yasunaga et al. (2018)	97.58
Xin et al. (2018)	97.58
Transformer-softmax (Guo et al., 2019)	97.04
BiLSTM-softmax (Yang et al., 2018)	97.51
BiLSTM-CRF (Yang et al., 2018)	97.51
BiLSTM-LAN	**97.65**

Table 6: Main results on WSJ.

As can be seen, a multi-layer model with larger hidden sizes does not give significantly better results compared to a 1-layer model with a hidden size of 400. We thus chose the latter for the final model.

For BiLSTM-LAN, each layer learns a more abstract representation of word and label distribution sequences. As shown in Table 2, for POS tagging, it is effective to capture label dependencies using two layers. More layers do not empirically improve the performance. We thus set the final number of layers to 3.

The effectiveness of Model Structure. To evaluate the effect of BiLSTM-LAN layers, we conduct ablation experiments as shown in Table 3. In BiLSTM-LAN w/o attention, we remove the attention inference sublayers from BiLSTM-LAN except for the last BiLSTM-LAN layer. This model is reminiscent to BiLSTM-softmax except that the output is based on label embeddings. It gives an accuracy slightly higher than that of LSTM-softmax, which demonstrates the advantage of label embeddings. On the other hand, it significantly underperforms BiLSTM-LAN (p-value<0.01), which shows the advantage of hierarchically-refined label distribution sequence encoding.

Model Size vs CRF. Table 2 also compares the effect of model sizes. We observe that: (1) As

		cs	da	en	fr	nl	no	pt	sv
BiLSTM-CRF	mean	98.42	95.77	95.41	96.94	94.65	97.07	97.78	96.06
	± std	0.03	0.12	0.06	0.08	0.11	0.11	0.04	0.07
(Yasunaga et al., 2018)	training(s)	268.74	18.17	58.20	70.10	44.49	56.06	51.59	15.54
BiLSTM-softmax	mean	98.48	95.90	95.36	97.01	94.76	97.26	97.78	95.98
	± std	0.04	0.09	0.17	0.09	0.17	0.03	0.05	0.08
	training(s)	129.14	9.27	25.02	33.65	23.00	28.84	23.13	8.16
BiLSTM-LAN	mean	**98.75**	**96.26**	**95.59**	**97.28**	**94.94**	**97.59**	**98.04**	**96.55**
	± std	0.02	0.12	0.13	0.08	0.11	0.04	0.04	0.01
	training(s)	165.64	11.32	33.04	40.48	29.71	37.06	27.48	10.40

Table 7: Multilingual POS tagging result on UD v2.2 treebanks, compared on 8 resource-rich languages.
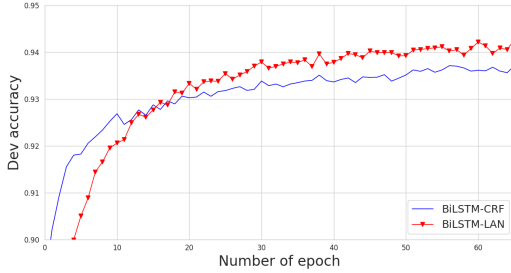

Figure 3: Training on the WSJ development set.

the model size increases, both BiLSTM-CRF and BiLSTM-LAN see a peak point beyond which further increase of model size does not bring better results, which is consistent with observations from prior work, demonstrating that the number of parameters is *not* the decisive factor to model accuracy; and (2) the best-performing BiLSTM-LAN model size is comparable to that of the BiLSTM-CRF model size, which indicates that the model structure is more important for the better accuracy of BiLSTM-LAN.

Speed vs CRF. Table 4 shows a comparison of training and decoding speeds. BiLSTM-LAN processes 805 and 714 sentences per second on the WSJ and CCGBank development data, respectively, outperforming BiLSTM-CRF by 3% and 19%, respectively. The larger speed improvement on CCGBank shows the benefit of lower asymptotic complexity, as discussed in Section 4.

Training vs CRF. Figure 3 presents the training curves on the WSJ development set. At the beginning, BiLSTM-LAN converges slower than BiLSTM-CRF, which is likely because BiLSTM-LAN has more complex layer structures for label embedding and attention. After around 15 training iterations, the accuracy of BiLSTM-LAN on the development sets becomes increasingly higher than BiLSTM-CRF. This demonstrates the effect of label embeddings, which allows more structured knowledge to be learned in modeling.

5.4 Final Results

WSJ. Table 5 shows the final POS tagging results on WSJ. Each experiment is repeated 5 times. BiLSTM-LAN gives significant accuracy improvements over both BiLSTM-CRF and BiLSTM-softmax ($p < 0.01$), which is consistent with observations on development experiments.

Table 6 compares our model with top-performing methods reported in the literature. In particular, Huang et al. (2015) use BiLSTM-CRF. Ma and Hovy (2016), Liu et al. (2017) and Yang et al. (2018) explore character level representations on BiLSTM-CRF. Zhang et al. (2018c) use S-LSTM-CRF, a graph recurrent network encoder. Yasunaga et al. (2018) demonstrate that adversarial training can improve the tagging accuracy. Xin et al. (2018) proposed a compositional character-to-word model combined with LSTM-CRF. BiLSTM-LAN gives highly competitive result on WSJ without training on external data.

Universal Dependencies(UD) v2.2. We design a multilingual experiment to compare BiLSTM-softmax, BiLSTM-CRF (strictly following Yasunaga et al. (2018) [1], which is the state-of-the-art on multi-lingual POS tagging) and BiLSTM-LAN. The accuracy and training speeds are shown in Table 7. Our model outperforms all the baselines on all the languages. The improvements are statistically significant for all the languages ($p < 0.01$), suggesting that BiLSTM-LAN is generally effective across languages.

OntoNotes 5.0. In NER, BiLSTM-CRF is widely used, because local dependencies between neighboring labels relatively more important that POS tagging and CCG supertagging. BiLSTM-LAN also significantly outperforms BiLSTM-CRF by 1.17 F1-score ($p < 0.01$). Table 8 compares BiLSTM-LAN to other published results on

---

[1] Note that our results are different from Table 2 of Yasunaga et al. (2018), since they reported results on UD v1.2

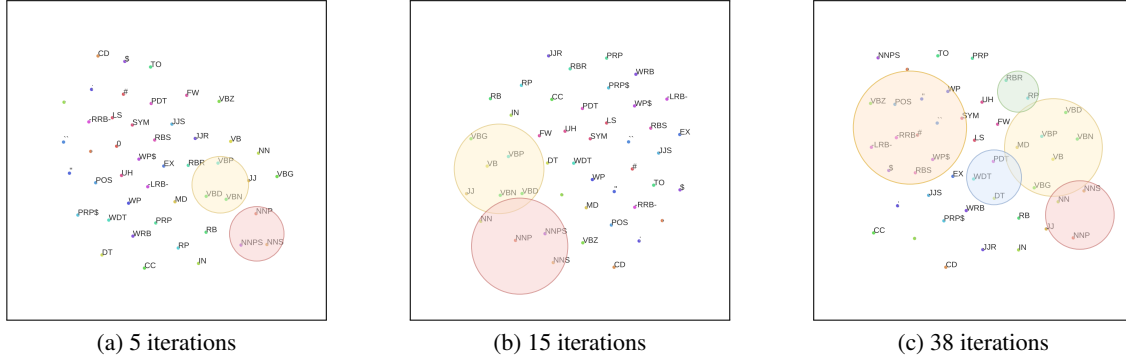| (a) 5 iterations | (b) 15 iterations | (c) 38 iterations |

Figure 4: t-SNE plot of label embeddings after different numbers of training iterations.

OntoNotes 5.0. Durrett and Klein (2014) propose a joint model for coreference resolution, entity linking and NER. Chiu and Nichols (2016) use a BiLSTM with CNN character encoding. Shen et al. (2017) introduce active learning to get better performance. Strubell et al. (2017) present an iterated dilated convolutions, which is a faster alternative to BiLSTM. Ghaddar and Langlais (2018) demonstrate that lexical features are actually quite useful for NER. Clark et al. (2018) present a cross-view training for neural sequence models. BiLSTM-LAN obtains highly competitive results compared with various of top-performance models without training on external data.

CCGBank. In CCG supertagging, the major challenge is a larger set of lexical tags $|L|$ and supertag constraints over long distance dependencies. As shown in Table 9, BiLSTM-LAN significantly outperforms both BiLSTM-softmax and BiLSTM-CRF ($p <0.01$), showing the advantage of LAN. Xu et al. (2015) and Vaswani et al. (2016a) explore BiRNN-softmax and BiLSTM-softmax, respectively. Søgaard and Goldberg (2016) present a multi-task learning architecture with BiRNN to improve the performance. Lewis et al. (2016) train BiLSTM-softmax using tri-training. Vaswani et al. (2016b) combine a LSTM language model and BiLSTM over the supertags. Tu and Gimpel (2019) introduce the inference network (Tu and Gimpel, 2018) in CCG supertagging to speed up the training and decoding for BiLSTM-CRF. Compared with these methods, BiLSTM-LAN obtains new state-of-the-art results on CCGBank, matching the tri-training performance of Lewis et al. (2016), without training on external data.

Model	F1
Durrett and Klein (2014)	84.04
Chiu and Nichols (2016)	86.28
Shen et al. (2017)	86.52
Strubell et al. (2017)	86.84
Ghaddar and Langlais (2018)	87.95
Clark et al. (2018)*	**88.81**
BiLSTM-softmax (Strubell et al., 2017)	83.76
BiLSTM-CRF (Strubell et al., 2017)	86.99
BiLSTM-LAN	88.16

Table 8: F1-scores on the OntoNotes 5.0 test set. * denotes semi-supervised and multi-task learning.

Model	Accuracy (%)
Xu et al. (2015)	93.0
Søgaard and Goldberg (2016)	93.3
Vaswani et al. (2016a)	94.2
Lewis et al. (2016)	94.3
Lewis et al. (2016)	94.7*
Vaswani et al. (2016b)	94.5
Tu and Gimpel (2019)	94.4
BiLSTM-softmax	94.1
BiLSTM-CRF	94.1
BiLSTM-LAN	**94.7**

Table 9: Supertagging accuracy on CCGbank test set. * indicates that further gains follow from semi-supervised tri-training (improving the accuracy from 94.3% to 94.7%).

6 Discussion

Visualization. A salient advantage of LAN is more interpretable models. We visualize the label embeddings as well as label attention weights for POS tagging. We use t-SNE to visualize the 45 different English POS tags (WSJ) on a 2D map after 5, 15, 38 training iteration, respectively. Each dot represents a label embedding. As can be seen in Figure 4, label embeddings are increasingly more meaningful during training. Initially, the vectors sit in random locations in the space. After 5 iterations, small clusters emerge, such as

Sentence	Gold Standard	LSTM-softmax	LSTM-CRF	LSTM-LAN
it	NP	NP	NP	NP
settled	(S[dcl]\NP)/PP	(S[dcl]\NP)/PP	(S[dcl]\NP)/PP	(S[dcl]\NP)/PP
with	((S\NP)\(S\NP))/NP	PP/NP	PP/NP	((S\NP)\(S\NP))/NP
a	NP[nb]/N	NP[nb]/N	NP[nb]/N	NP[nb]/N
loss	N	N	N	N
of	(NP\NP)/NP	(NP\NP)/NP	(NP\NP)/NP	(NP\NP)/NP
4.95	N / N	N / N	N / N	N / N
cents	N	N	N	N
at	PP/NP	(NP\NP)/NP	PP/NP	PP/NP
$	N / N[num]	N / N[num]	N / N[num]	N / N[num]
1.3210	N[num]	N[num]	N[num]	N[num]
a	(NP\NP) / N	(NP\NP) / N	(NP\NP) / N	(NP\NP) / N
pound	N	N	N	N
.	.	.	.	.

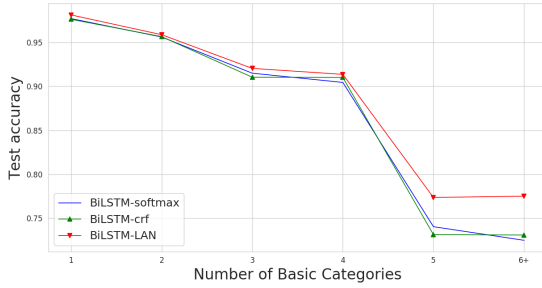Table 10: CCG case analysis. The error are in yellow.

Figure 5: Accuracy against supercategory complexity.

"NNP" and "NNPS", "VBD" and "VBN", "JJS" and "JJR" etc. The clusters grow absorbing more related tags after more training iterations. After 38 training iterations, most similar POS tags are grouped together, such as "VB", "VBD", "VBN", "VBG" and "VBP". More attention visualization are shown in Appendix B.

Supercategory Complexity. We also measure the complexity of supercategories by the number of basic categories that they contain. According to this definition, "S", "S/NP" and "(S \NP)/NP" have complexities of 1, 2 and 3, respectively. Figure 5 shows the accuracy of BiLSTM-softmax, BiLSTM-CRF and BiLSTM-LAN against the supertag complexity. As the complexity increases, the performance of all the models decrease, which conforms to intuition. BiLSTM-CRF does not show obvious advantages over BiLSTM-softmax on complex categories. In contrast, BiLSTM-LAN outperforms both models on complex categories, demonstrating its advantage in capturing more sophisticated label dependencies.

Case Study. Some predictions of BiLSTM-softmax, BiLSTM-CRF and BiLSTM-LAN are shown in Table 10. The sentence contains two prepositional phrases "with ..." and "at ...", thus exemplifies the PP-attachment problem, one of the hardest sub-problems in CCG supertagging. As can be seen, BiLSTM-softmax fails to learn the long-range relation between "settled" and "at". BiLSTM-CRF strictly follows the hard constraint between neighbor categories thanks to Markov label transition. However, it predicts "with" incorrectly as "PP/NP" with the former supertag ending with "/PP". In contrast, BiLSTM-LAN can capture potential long-term dependency and better determine the supertags based on global label information. In this case, our model can effectively represent a full label search space without making Markov assumptions.

7 Conclusion

We investigate a hierarchically-refined label attention network (LAN) for sequence labeling, which leverages label embeddings and captures potential long-range label dependencies by deep attentional encoding of label distribution sequences. Both in theory and empirical results prove that BiLSTM-LAN effective solve label bias issue. Results on POS tagging, NER and CCG supertagging show that BiLSTM-LAN outperforms BiLSTM-CRF and BiLSTM-softmax.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 452–461. Association for Computational Linguistics.

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

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925, Brussels, Belgium. Association for Computational Linguistics.

Cícero Nogueira Dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1818–II–1826. JMLR.org.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490.

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

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2019. Star-transformer. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1315–1325, Minneapolis, Minnesota. Association for Computational Linguistics.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1693–1701, Cambridge, MA, USA. MIT Press.

Julia Hockenmaier and Mark Steedman. 2007. Ccgbank: A corpus of ccg derivations and dependency structures extracted from the penn treebank. *Comput. Linguist.*, 33(3):355–396.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 57–60, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *CoRR*, abs/1508.01991.

Rudolf Kadlec, Martin Schmid, Ondřej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 908–918. Association for Computational Linguistics.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *TACL*, 4:313–327.

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

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

Mike Lewis, Kenton Lee, and Luke Zettlemoyer. 2016. Lstm ccg parsing. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 221–231. Association for Computational Linguistics.

Qi Li, Tianshi Li, and Baobao Chang. 2016. Discourse parsing with attention-based hierarchical neural networks. In *Proceedings of the 2016 Conference on*

Empirical Methods in Natural Language Processing, pages 362–371. Association for Computational Linguistics.

Jiangming Liu and Yue Zhang. 2017. Attention modeling for targeted sentiment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 572–577. Association for Computational Linguistics.

Liyuan Liu, Jingbo Shang, Frank Xu, Xiang Ren, Huan Gui, Jian Peng, and Jiawei Han. 2017. Empower sequence labeling with task-aware neural language model. *arXiv preprint arXiv:1709.04109*.

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

Christopher D. Manning. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *Computational Linguistics and Intelligent Text Processing*, pages 171–189, Berlin, Heidelberg. Springer Berlin Heidelberg.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330.

Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. 2016. All-in text: Learning document, label, and word representations jointly. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 1948–1954. AAAI Press.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, John Bauer, Sandra Bellato, Kepa Bengoetxea, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Radu Ion, Elena Irimia, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Tolga Kayadelen, Václava Kettnerová, Jesse Kirchner, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phng Lê H`ông, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Shinsuke Mori, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lng Nguyễn Thị, Huy`ên Nguyễn Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adédayọ̀ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roca, Olga Rudina, Shoval Sadde, Shadi Saleh, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Yuta Takahashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan

van Noord, Viktor Varga, Veronika Vincze, Lars Wallin, Jonathan North Washington, Seyi Williams, Mats Wirén, Tsegay Woldemariam, Tak-sum Wong, Chunxiao Yan, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. 2018. Universal dependencies 2.2. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. 2009. Zero-shot learning with semantic output codes. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1410–1418. Curran Associates, Inc.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418. Association for Computational Linguistics.

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

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2017. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389. Association for Computational Linguistics.

Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep active learning for named entity recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 252–256, Vancouver, Canada. Association for Computational Linguistics.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. 2013. Zero-shot learning through cross-modal transfer. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, pages 935–943, USA. Curran Associates Inc.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038. Association for Computational Linguistics.

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

Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1165–1174. ACM.

Zhiyang Teng and Yue Zhang. 2018. Two local models for neural constituent parsing. In *Proceedings*

of the 27th International Conference on Computational Linguistics, pages 119–132. Association for Computational Linguistics.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lifu Tu and Kevin Gimpel. 2018. Learning approximate inference networks for structured prediction. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Lifu Tu and Kevin Gimpel. 2019. Benchmarking approximate inference methods for neural structured prediction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3313–3324, Minneapolis, Minnesota. Association for Computational Linguistics.

A. Vaswani, Y. Bisk, K. Sagae, and R. Musa. 2016a. Supertagging with LSTMs. In *Proc. NAACL*.

Ashish Vaswani, Yonatan Bisk, Kenji Sagae, and Ryan Musa. 2016b. Supertagging with lstms. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 232–237. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018a. Joint embedding of words and labels for text classification. *arXiv preprint arXiv:1805.04174*.

Lu Wang, Shoushan Li, Changlong Sun, Luo Si, Xiaozhong Liu, Min Zhang, and Guodong Zhou. 2018b. One vs. many qa matching with both word-level and sentence-level attention network. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2540–2550. Association for Computational Linguistics.

Xun Wang, Katsuhito Sudoh, and Masaaki Nagata. 2015. Empty category detection with joint context-label embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 263–271. Association for Computational Linguistics.

Wei Wu, Houfeng Wang, Tianyu Liu, and Shuming Ma. 2018. Phrase-level self-attention networks for universal sentence encoding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3729–3738. Association for Computational Linguistics.

Yingwei Xin, Ethan Hart, Vibhuti Mahajan, and Jean-David Ruvini. 2018. Learning better internal structure of words for sequence labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2584–2593, Brussels, Belgium. Association for Computational Linguistics.

Chang Xu, Cecile Paris, Surya Nepal, and Ross Sparks. 2018. Cross-target stance classification with self-attention networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 778–783. Association for Computational Linguistics.

Wenduan Xu, Michael Auli, and Stephen Clark. 2015. Ccg supertagging with a recurrent neural network. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 250–255. Association for Computational Linguistics.

Jie Yang, Shuailong Liang, and Yue Zhang. 2018. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*.

Jie Yang and Yue Zhang. 2018. Ncrf++: An open-source neural sequence labeling toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Michihiro Yasunaga, Jungo Kasai, and Dragomir Radev. 2018. Robust multilingual part-of-speech tagging via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 976–986. Association for Computational Linguistics.

Honglun Zhang, Liqiang Xiao, Wenqing Chen, Yongkun Wang, and Yaohui Jin. 2018a. Multi-task label embedding for text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4545–4553. Association for Computational Linguistics.

Yuan Zhang, Hongshen Chen, Yihong Zhao, Qun Liu, and Dawei Yin. 2018b. Learning tag dependencies

for sequence tagging. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4581–4587. International Joint Conferences on Artificial Intelligence Organization.

Yue Zhang, Qi Liu, and Linfeng Song. 2018c. Sentence-state lstm for text representation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 317–327. Association for Computational Linguistics.

Z. Zhang and V. Saligrama. 2016. Zero-shot learning via joint latent similarity embedding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6034–6042.
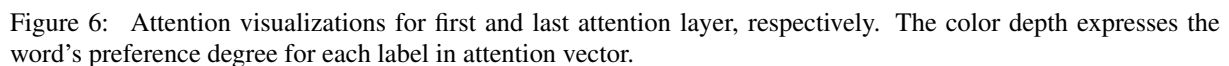
Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1127–1137. Association for Computational Linguistics.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212. Association for Computational Linguistics.

Appendices

A Hyper-Parameters

Hyper-parameter	Value
character embeddings	30
character-level hidden size	50
word-level hidden size (English)	400
word-level hidden size (other languages)	200
encoder layers (POS)	3
encoder layers (NER)	4
encoder layers (CCG)	5
number of attention heads	5
drop out	0.5
batch size	10
optimizer	SGD
momentum	0.9
L2 regularization	$1e^{-8}$
learning rate	0.01
decay rate (POS)	0.035
decay rate (NER)	0.04
decay rate (CCG)	0.05
gradient clipping	5.0

Table 11: Hyper-parameters.

Tabel 11 shows the hyper-parameters of BiLSTM-LAN used for all the experiments in the papers.

B Attention Visualization

Figure 6: Attention visualizations for first and last attention layer, respectively. The color depth expresses the word's preference degree for each label in attention vector.

Here we continue the analysis of visualization from Section 6, we also visualize the label attention weights of BiLSTM-LAN, given the sentence in "Tell (VB) us (PRP) about (IN) spending (NN) restraint (NN)". As shown in Figure 6, the first layer contains initial label distributions according to unigram information, in which much ambiguity remains. For example, likely POS tags for the word "*spending*" include "JJ", "NN", "VBG", "TO" and "RBS", while the attention layer assigns some probabilities to almost every label. In the second layer, the label distribution of every word becomes sharp and concentrated on the most probable tags in the sentential context. Here the word "spending" bares only "NN" and "VBG" labels, with relatively more probability (0.114) being assigned to "NN".