The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)

Encoder-Decoder Based Unified Semantic Role Labeling with Label-Aware Syntax

Hao Fei, Fei Li, Bobo Li, Donghong Ji*

Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China

{hao.fei, boboli, dhji}@whu.edu.cn, foxlf823@gmail.com

Abstract

Currently the unified semantic role labeling (SRL) that achieves predicate identification and argument role labeling in an end-to-end manner has received growing interests. Recent works show that leveraging the syntax knowledge significantly enhances the SRL performances. In this paper, we investigate a novel unified SRL framework based on the sequence-to-sequence architecture with double enhancement in both the encoder and decoder sides. In the encoder side, we propose a novel label-aware graph convolutional network (LA-GCN) to encode both the syntactic dependent arcs and labels into BERT-based word representations. In the decoder side, we creatively design a pointer-network-based model for detecting predicates, arguments and roles jointly. Our pointer-net decoder is able to make decisions by consulting all the input elements in a global view, and meanwhile it is syntactic-aware by incorporating the syntax information from LA-GCN. Besides, a high-order interacted attention is introduced into the decoder for leveraging previously recognized triplets to help the current decision. Empirical experiments show that our framework significantly outperforms all existing graph-based methods on the CoNLL09 and Universal Proposition Bank datasets. In-depth analysis demonstrates that our model can effectively capture the correlations between syntactic and SRL structures.

1 Introduction

Semantic role labeling (SRL), as a shallow semantic parsing for extracting the *predicate–argument* structure in sentences, has long been a fundamental natural language processing (NLP) task (Gildea and Jurafsky 2000; Pradhan et al. 2005; Zhao et al. 2009; Fei, Ren, and Ji 2020). Traditional SRL is divided into two pipeline subtasks: predicate identification (Scheible 2010) and argument role labeling (Pradhan et al. 2005). Recently, great efforts have been paid for constructing unified SRL systems, solving two pipeline steps in one shot via one unified model (He et al. 2018), as illustrated in Figure 1. Most current unified SRL models employ graph-based models by exhaustively enumerating all potential predicates and their corresponding arguments jointly (He et al. 2018; Cai et al. 2018; Li et al. 2019). Graph-based models improve the SRL performances by reducing the error propagation in the pipeline scheme, meanwhile achieving an overall simplified procedure.

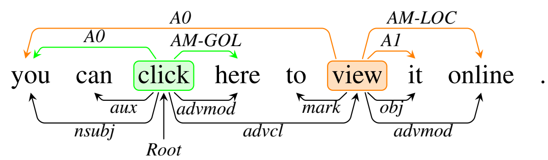

Figure 1: An example of unified SRL. Above/below are the SRL/dependency structures, respectively. Same color refers to the propositions under the same predicate.

Moreover, many prior work has revealed that syntax features are extraordinarily effective for SRL (Roth and Lapata 2016; Marcheggiani and Titov 2017; Zhang, Wang, and Si 2019). Intuitively, the *predicate–argument* structure in SRL shares many common connections with the underlying syntactic structure (i.e., dependency tree), as illustrated in Figure 1. Nevertheless, most of prior syntactic dependency-aware SRL models are limited to the merely use of dependency arcs (e.g., click⌒you), while neglecting dependency labels (e.g., *nsubj*) (Jin, Kawahara, and Kurohashi 2015; Roth and Lapata 2016; Strubell et al. 2018; Xia et al. 2019). Actually, dependency labels carry crucial evidences for SRL, since the information from the neighboring nodes under distinct types of arcs contributes in different degrees (Kasai et al. 2019). In Figure 1, dependent arcs with nominal attributes (e.g., *nsubj*, *obj*) are more associated with core roles (e.g., `A0`-`A5`) in SRL, while the arcs with modifying attributes (e.g., *advmod*) relate more to modifier roles (e.g., `AM-*`). On this basis, another line of works has considered both dependent arcs and labels via graph convolutional networks (GCN) (Marcheggiani and Titov 2017).

In this work, we propose a novel unified SRL framework, which is fully orthogonal to the existing graph-based methods. The system leverages the encoder-decoder architecture (cf. Figure 2), being able to jointly produce all the possible *predicate-argument-role* triplets. In the encoder side, we newly propose a label-aware GCN (namely LA-GCN) for encoding word representations, dependency arcs and labels (cf. Figure 2 left). The syntax GCN encoders in existing SRL works (Marcheggiani and Titov 2017) model depen-

*Corresponding author

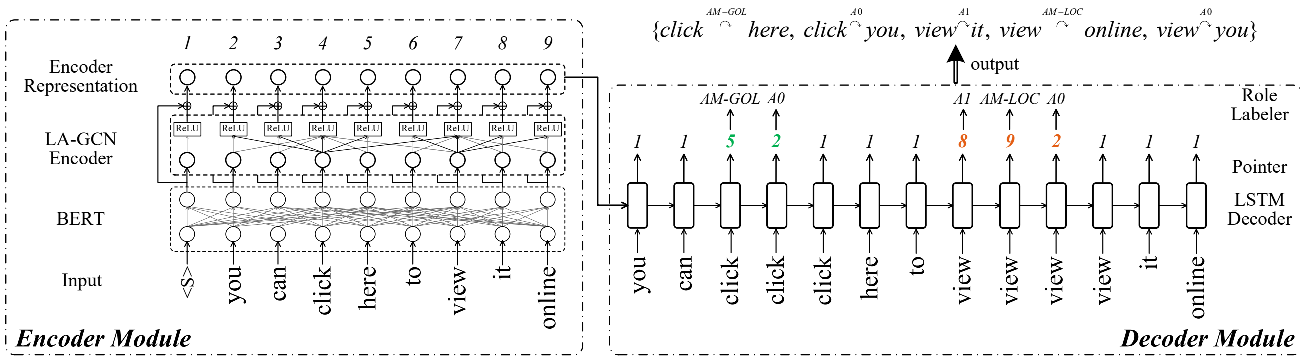

Figure 2: Our sequence-to-sequence framework for unified SRL. The input consists of a sentinel token '<S>' (idx=1) and the words (idx=2∼9). During decoding, if the pointer directs to <S>, the decoder will go to next word. If the pointer directs to a real word, a predicate-argument pair will be generated and its role label will be given by the role labeler.

dent labels by creating label-specific GCN parameters, i.e, each dependency label owns a set of GCN parameters. Such an implicit modeling method could be inefficient and problematic to be re-harnessed, while our LA-GCN encoder formalizes the arcs and labels simultaneously, and normalizes them unitedly into distributions as connecting-strengths (cf. Figure 3).

In the decoder side, we design a novel decoding system for SRL based on the pointer network (Vinyals, Fortunato, and Jaitly 2015), which picks an element from the input tokens with the highest probability at each decoding step (cf. Figure 2 right). Pointer networks have been exploited for a wide range of NLP tasks, such as text summarization (See, Liu, and Manning 2017), syntactic parsing (Ma et al. 2018), information extraction (Li, Ye, and Shang 2019), etc. Pointer networks can make decisions in the consultant of all the input elements in a global scope (See, Liu, and Manning 2017; Ma et al. 2018; Li, Ye, and Shang 2019). The decoder detects all possible arguments for each predicate in an incremental, close-first fashion, and meanwhile the labeler assigns semantic roles for the determined *argument-predicate* pairs. Besides the pointer-based decoding, we further propose two enhancements in the decoder side. First, we improve the pointer with a high-order interacted attention mechanism, where each decoding input is fused with the prior recognized triplets as high-order information (cf. Figure 4). At the meantime, we render the pointer to be syntax-aware by fully incorporating previously yielded dependency information from the LA-GCN (cf. Figure 4).

We conduct evaluations on two SRL benchmark datasets, including CoNLL09 English (Hajič et al. 2009), and Universal Proposition Bank (Akbik et al. 2015; Akbik and Li 2016) for total eight languages. Experimental results show that our framework outperforms all baselines significantly, and achieves new state-of-the-art performances on unified SRL in terms of both the predicate identification and argument role labeling. Ablation studies are performed for comprehensively understanding the contribution of each proposed mechanism. Further analysis demonstrates that our model can effectively capture the correlations between dependent labels and semantic role labels. At last, we summarize our contributions in this study as below:

▶ To our knowledge, we are the first to present a unified SRL framework based on pointer networks. The pointer at current decision is encouraged to interact with previously recognized triplets via a proposed high-order interacted attention mechanism.

▶ We introduce a novel label-aware GCN encoder for modeling syntactic dependency arcs and labels simultaneously. We render the pointer to be syntax-aware by reharnessing the syntactic dependency distribution from the label-aware GCN.

▶ Our framework gives new state-of-the-art performances on SRL benchmarks. In-depth analysis uncovers that our model can well capture the correlations between syntactic and SRL structures.

2 Related Work

The task of semantic role labeling (SRL) pioneered by Gildea and Jurafsky (2000) can be roughly grouped into two schemes: the pipeline scheme and the unified scheme. Prior works traditionally separate the SRL into two pipeline subtasks, i.e., predicate disambiguation and argument role labeling. They mainly conduct argument role labeling based on the pre-identified predicate oracle. Earlier works employ hand-crafted features with machine learning classifiers (Pradhan et al. 2005; Punyakanok, Roth, and Yih 2008), while later researchers take advantages of neural networks with automatic distributed features (FitzGerald et al. 2015; Roth and Lapata 2016; Marcheggiani and Titov 2017; Strubell et al. 2018). Recent efforts are paid for the end-to-end solution that handles both two subtasks by one model, i.e., unified SRL. All the current unified SRL employs graph-based neural model, exhaustively enumerating all the possible predicate and arguments, as well as the roles labels (He et al. 2018; Cai et al. 2018; Li et al. 2019; Lei et al. 2015).

Our work is also closely related to the application of pointer networks (Vinyals, Fortunato, and Jaitly 2015). Pointer networks have been extensively exploited for many NLP tasks, such as text summarization (See, Liu, and Manning 2017), syntactic parsing (Ma et al. 2018; Fernández-

González and Gómez-Rodríguez 2020), information extraction (Li, Ye, and Shang 2019), etc. On the one hand, based on encoder-decoder architecture (Sutskever, Vinyals, and Le 2014), pointer network is able to yield results with lower model complexity. Besides, pointer network can make decisions in the consultant of all the input elements in global viewpoint (See, Liu, and Manning 2017; Ma et al. 2018; Li, Ye, and Shang 2019). In this work, we first present a pointer network based solution for end-to-end SRL, as an alternative to the current graph-based methods.

In addition, our work also falls into syntax-aware SRL. The syntactic features, e.g., dependency structures, have long been proven effective for SRL tasks (Marcheggiani, Frolov, and Titov 2017; Zhang, Wang, and Si 2019; Fei et al. 2020). Yet most prior work merely utilized the dependent arcs, leaving the dependent type information unexploited, which however is equally crucial for the task. On the other hand, some consider encoding such dependent labels using GCN model (Marcheggiani and Titov 2017), i.e., by creating the label-specific GCN parameters. Unfortunately, such implicit integration of dependent labels could be less effective, since intuitively the dependent arcs and labels are closely related, and should be modeled in a more unified manner. In this paper, we propose a label-aware GCN encoder for effectively modeling these two syntax information. We also improve our framework with a high-order interacted attention mechanism and a syntactic-aware pointer mechanism, which can further facilitate the procedure.

3 Preliminary

SRL Formalization Following the current line of unified SRL studies (He et al. 2018; Cai et al. 2018; Li et al. 2019), we model the task as predicate-argument-role triplet prediction. Given an input sentence $S = \{w_1, \cdots, w_n\}$, our framework predicts a set of triplets $Y = \{\cdots, <p_k, a_k, r_k>, \cdots | p_k \in P, a_k \in A, r_k \in R\}$, where P, A and R are all possible predicate/argument tokens, and role labels. In Figure 2, we show the format of the final outputs for the corresponding input sentence. In this work, we mainly consider the dependency-based SRL, detecting the dependency head of each argument (Li et al. 2019). Note that our framework is flexible to be extended to the span-based SRL, by further predicting the end-boundary for an argument.

Pointer Network The pointer mechanism functions by learning the conditional probability of an output at the position corresponding to an input token. The vanilla pointer network employ the attention mechanism (Bahdanau, Cho, and Bengio 2015; Luong, Pham, and Manning 2015) to accomplish the selecting target from the input sequence.[1] Technically, given the encoding representations of input tokens $[\boldsymbol{e}_1, \cdots, \boldsymbol{e}_n]$, and the current decoding representation $\boldsymbol{s}_t$, we calculate the relatedness between $\boldsymbol{s}_t$ and each $\boldsymbol{e}_i$ and normalize it over the inputs as below:

$$\begin{aligned} v_{t,i} &= \text{Score}(\boldsymbol{s}_t, \boldsymbol{e}_i) = \text{Tanh}(\boldsymbol{W}_0[\boldsymbol{s}_t; \boldsymbol{e}_i] + b)\,, \\ o_{t,i} &= \text{Softmax}(v_{t,i})\,, \quad i = [1, \cdots, n]\,. \end{aligned} \tag{1}$$

[1] We denote the vanilla pointer mechanism as *base pointer*.

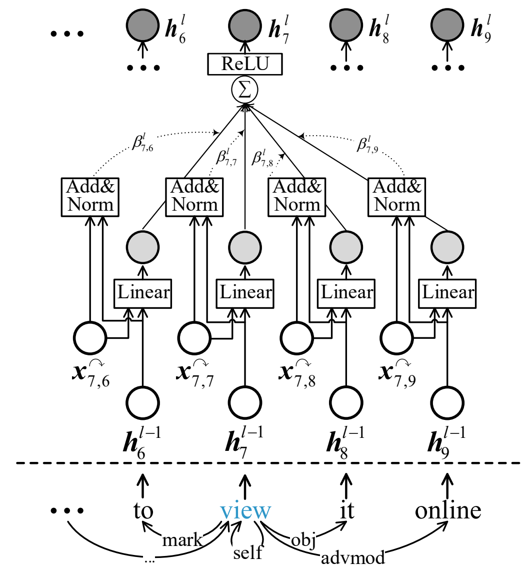

Figure 3: Illustration to show the process of generating the encoder representation $\boldsymbol{h}_7$ for the 7-th word ‘view’ in Figure 2, using our proposed label-aware GCN.

We then take the position P_t with the maximum $o_{t,i}$ over each input token w_i as the output of t-th decoding:

$$P_t = \underset{i}{\text{Argmax}}(o_{t,i})\,. \tag{2}$$

Each pointing decision is made in the consultant of all input tokens, which can be seen as a type of global interaction.

4 Our Framework for Unified SRL

Overview In Figure 2, we give a overall picture of our framework. In the encoder module, we first employ the pre-trained BERT language model (Devlin et al. 2019) to provide word representations for input tokens. Then, the LA-GCN fuses rich syntax features and learn contextualized word representations. Finally, we take the shortcut connected representations between BERT representations and LA-GCN representations as the final encoder representations. As shown in Figure 2, a sentinel token ‘<S>’ is inserted at the head of the input sentence, which informs the system that *the current decoding input (maybe predicate) has no further argument*.

In the decoder module, the decoder first takes the encoder representation $\boldsymbol{e}_j$ of the input token w_j, and outputs the corresponding decoding representation $\boldsymbol{s}_t$ at each step t. Then the pointer will direct to a position of a input token w_i, and if w_i is not ‘<S>’, the decoder will generate a *predicate*(w_j)-*argument*(w_i) pair. Further, the labeler assigns the role label for the pair. Note that the decoder takes tokens sequentially in the order of the input, but if a token w_j is determined as a predicate, it will be re-input at next decoding step, until no further argument is detected. The decoding process terminates once all the input tokens are examined, and meantime it outputs all the triplets Y.

Encoder Module

Word Representation We employ BERT (Vaswani et al. 2017) to yield our contextualized word representation $\boldsymbol{r}_i^b$ for each token w_i:

$$\{\boldsymbol{r}_1^b, \cdots, \boldsymbol{r}_n^b\} = \text{BERT}(\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}), \tag{3}$$

where $\boldsymbol{r}_i^b$ is the BERT output representation. We concatenate word-piece embeddings and position embeddings, i.e., $\boldsymbol{x}_i = \boldsymbol{x}_i^w \oplus \boldsymbol{x}_i^P$, as the BERT input embeddings.

Label-Aware GCN We propose a label-aware GCN (LA-GCN) encoder for simultaneously modeling the dependency arcs and labels (Figure 3). We denote $a_{i,j}$=1 if there is an arc between w_i and w_j, and $a_{i,j}$=0 vice versa. $\pi_{i,j}^{\frown}$ represents the dependency label between w_i and w_j. Besides of the predefined dependency labels, we additionally add a '*self*' label as the self-loop arc $\pi_{i,i}^{\frown}$ for w_i, and a '*none*' label for representing no arc between w_i and w_j. We use the embedding $\boldsymbol{x}_{i,j}^{\frown}$ for the label $\pi_{i,j}^{\frown}$. Our LA-GCN consists of L layers, and we denote the resulting hidden representation of w_i at l-th layer as $\boldsymbol{h}_i^{(l)}$:

$$\boldsymbol{h}_i^{(l)} = \text{ReLU}(\textstyle\sum_{j=1}^n \beta_{i,j}^{(l)} (\boldsymbol{W}_1 \cdot \boldsymbol{h}_j^{(l-1)} + \boldsymbol{W}_2 \cdot \boldsymbol{x}_{i,j}^{\frown} + b)) \,, \tag{4}$$

where $\beta_{i,j}^{(l)}$ is the neighbor connecting-strength distribution:

$$\beta_{i,j}^{(l)} = \frac{a_{i,j} \cdot \exp\left(\boldsymbol{r}_i^n \cdot \boldsymbol{r}_j^n\right)}{\sum_{j=1}^n a_{i,j} \cdot \exp\left(\boldsymbol{r}_i^n \cdot \boldsymbol{r}_j^n\right)} \,, \tag{5}$$

where $\boldsymbol{r}_i^n$ is the element-wise addition by $\boldsymbol{h}_i^{(l-1)} + \boldsymbol{x}_{i,j}^{\frown}$. The neighbor connecting-strength distribution $\beta_{i,j}$ encodes both the information from the dependent arcs and labels, and thus comprehensively reflects the syntactic attributes. The initial input representation of the first layer LA-GCN is $\boldsymbol{h}_i^{(0)} = \boldsymbol{r}_i^b \oplus \boldsymbol{x}_i^{pos}$, where $\boldsymbol{x}_i^{pos}$ is the POS tag embedding for w_i. We note that after L layers of information aggregation, much high-order information between neighbors in the graph can fully retained, which will facilitate the SRL.

We then impose a residual connection between BERT and the last layer of LA-GCN. The final encoder representations are the concatenation of the two representations from two encoders respectively, which ensures the minimum information loss from the BERT contextualized source.

$$\boldsymbol{e}_i = \boldsymbol{h}_i^{(L)} \oplus \boldsymbol{r}_i^b \,. \tag{6}$$

Decoder Module

We employ the LSTM (Hochreiter and Schmidhuber 1997) as our decoder. Suppose the system has already confirmed several triplets $Y' = \{\cdots, <p_k, a_k, r_k>\}$, with the corresponding representations $\boldsymbol{r}_k^{trpl}$ for these established triplets (elaborated later in Eq. 12), at the decoding step t. The decoding input representation is $\boldsymbol{u}_t = \boldsymbol{e}_\star \oplus \boldsymbol{e}_\star^\dagger$, where $\boldsymbol{e}_\star$ is the corresponding encoder representation of the $\star$-th input token (from Eq. 6), and $\boldsymbol{e}_\star^\dagger$ is a high-order representation via the high-order interacted attention that we will introduce in the next sub-section. Then, the LSTM cell produces the decoding representation based on $\boldsymbol{u}_t$:

$$\boldsymbol{s}_t = \text{LSTMCell}(\boldsymbol{u}_t) \,. \tag{7}$$

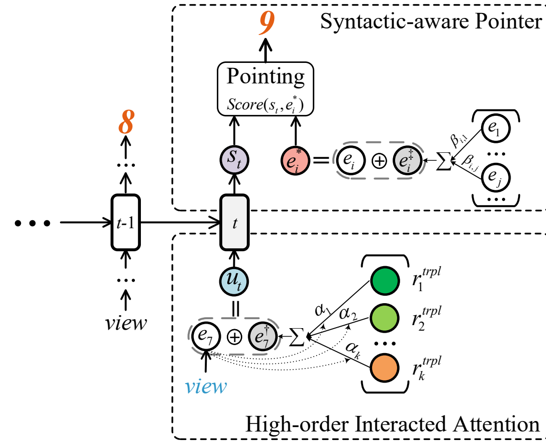

Figure 4: Illustration to show the input and output for the decoding step at 'view' in Figure 2. t →*the current decoding step*, i →*the index of an input word*, 7 →*the index of the current input word*, r_k^{trpl} →*previously recognized triplet representations*.

High-Order Interacted Attention The representation $\boldsymbol{e}_\star^\dagger$ from the high-order interacted attention can be calculated as:

$$\begin{aligned} \boldsymbol{e}_\star^\dagger &= \textstyle\sum_{j=1}^k \alpha_j \boldsymbol{r}_j^{trpl} \,, \\ \alpha_j &= \text{Softmax}(\mu_j) \,, \\ \mu_j &= \text{Tanh}(\boldsymbol{W}_3 \boldsymbol{r}_j^{trpl} + \boldsymbol{W}_4 \boldsymbol{e}_\star + b) \,. \end{aligned} \tag{8}$$

By interacting with previously recognized triplets, the current decision will capture the higher-order information. Otherwise, the decoding process for detecting the argument for a predicate is restricted to local features, which can be seen as a first-order model. Taking the example in Figure 4, for the predicate '*view*', being informed by the prior triplet, e.g., '*view-it-A1*', the pointer tends to assign other token as argument rather than '*it*' repeatedly.

Syntax-Aware Pointer Mechanism Based on the decoding representation $\boldsymbol{s}_t$, we can obtain the pointing result via the *base pointer* in Eq. (1). We now consider making full use of the previously yielded syntactic information in LA-GCN, i.e., rendering the pointer to be syntactic-aware. Specifically, for each encoder representation $\boldsymbol{e}_i$, we calculate its corresponding syntactic-aware counterpart (denoted as $\boldsymbol{e}_i^\ddagger$):

$$\boldsymbol{e}_i^\ddagger = \textstyle\sum_{j \neq i} \beta_{i,j}^{(L)} \boldsymbol{e}_j \,, \tag{9}$$

where $\beta_{i,j}^{(L)}$ is the syntax connecting distribution at the last layer of LA-GCN. We then concatenate $\boldsymbol{e}_i^\ddagger$ and $\boldsymbol{e}_i$ as the syntax-aware representation $\boldsymbol{e}_i^*$, which together with $\boldsymbol{s}_t$ is fed into the pointer in Eq. (1) for producing the pointing position. The reuse of $\beta_{i,j}$ can be understood as a way to further leverage the second-order syntactic structures concerning the current $\boldsymbol{e}_i$ for the pointer.

Role Labeler Once a predicate-argument pair has been determined, the role labler will assign a role for the

			In-domain (*WSJ*)				Out-of-domain (*Brown*)			
			P	R	F1	Prd	P	R	F1	Prd
Without BERT	He et al. (2018)	`LP-GCN`	89.4	88.7	89.4	93.1	79.8	78.1	79.0	79.6
	Cai et al. (2018)	`LP-GCN`	90.2	89.3	89.9	94.9	80.4	78.9	79.5	80.5
		`GCN*`	89.7	89.0	89.5	94.2	80.0	78.3	79.0	80.0
	Li et al. (2019)	`LP-GCN`	90.2	89.5	89.9	95.2	79.9	78.5	79.2	81.5
		`GCN*`	89.8	89.1	89.4	94.7	79.4	78.2	78.7	80.7
	Ours	`LA-GCN`	**90.8**	**90.0**	**90.4**	**95.5**	**80.7**	**79.3**	**80.2**	**81.8**
		`LP-GCN`	90.6	89.6	90.2	95.4	80.5	79.0	79.7	81.6
		`GCN*`	89.6	89.0	89.3	94.5	79.0	78.3	78.6	80.2
With BERT	Cai et al. (2018)	`LP-GCN`	91.2	92.0	91.7	95.7	82.1	82.8	83.9	85.6
		`LA-GCN`	91.0	92.3	91.9	95.8	82.5	83.1	84.3	85.8
	Li et al. (2019)	`LP-GCN`	91.6	92.2	92.0	95.5	84.7	84.4	84.6	87.0
		`LA-GCN`	92.0	92.4	92.2	95.9	85.4	85.1	85.2	87.8
	Ours	`LA-GCN`	**92.5**	**92.5**	**92.5**	**96.2**	**85.6**	**85.3**	**85.4**	**90.0**

Table 1: Unified SRL results on CoNLL09 English data.

predicate-argument pair via a Biaffine layer (Dozat and Manning 2017):

$$\begin{aligned} \boldsymbol{r}^{lb} &= g_1(\boldsymbol{r}^p)\boldsymbol{W}_5 g_2(\boldsymbol{r}^a) + \boldsymbol{U}_1\boldsymbol{r}^p + \boldsymbol{U}_2\boldsymbol{r}^a + b\,, \\ p_r &= \text{Softmax}(\boldsymbol{r}^{lb})\,, \end{aligned} \tag{10}$$

where $g_1(\cdot)$ and $g_2(\cdot)$ are two feedforward layers, $\boldsymbol{r}^p$ and $\boldsymbol{r}^a$ are the predicate and argument representations, given by:

$$\boldsymbol{r}^p = \boldsymbol{e}^p \oplus \boldsymbol{s}\,, \quad \boldsymbol{r}^a = \boldsymbol{e}^a \oplus \boldsymbol{s}\,, \tag{11}$$

where $\boldsymbol{e}^p$ and $\boldsymbol{e}^a$ are the corresponding encoder representations (cf. Eq. 6) of the predicate and argument, and $\boldsymbol{s}$ is the decoding representation (cf. Eq. 7). For each established triplet $<p_k, a_k, r_k>$, we construct the corresponding representations $\boldsymbol{r}_k^{trpl}$ by applying element-wise addition on the representations $\boldsymbol{r}_k^a$, $\boldsymbol{r}_k^p$ and $\boldsymbol{r}_k^{lb}$ of the corresponding predicate, argument and role:

$$\boldsymbol{r}_k^{trpl} = \boldsymbol{r}_k^a + \boldsymbol{r}_k^p + \boldsymbol{r}_k^{lb}\,, \tag{12}$$

where $\boldsymbol{r}_k^{trpl}$ is used in the high-order interacted attention.

Learning

As described earlier, each decoding input in our framework depends on the last prediction. Following previous encoder-decoder studies, we adopt the teacher-forcing strategy (Williams and Zipser 1989) during training, maximizing the likelihood of the oracle pointer at each decoding frame. Besides, we change the order of oracles during learning so that the process of detecting the arguments for a predicate is in a close-first manner. For example, for the predicate '*view*' in Figure 1, the outputs will follow the from-near-to-far order: '*it*→*online*→*you*', instead of '*you*→*it*→*online*' or other cases. This is more intuitive compared with the traditional reading order (from-left-to-right), as humans always tend to first grasp the core ideas then dig into more details. Also this facilitates the high-order interacted attention mechanism.

The training target is to minimize the cross-entropy loss between the predicted pointer distribution and the gold one:

$$\mathcal{L}_{pointer} = -\textstyle\sum_t^N \hat{\boldsymbol{o}}_t \log \boldsymbol{o}_t\,, \tag{13}$$

where N is the total decoding steps and $\hat{\boldsymbol{o}}_t$ is the gold pointing. Also there is the role label loss:

$$\mathcal{L}_{role} = -\textstyle\sum_k^M \hat{p}_k \log p_k\,, \tag{14}$$

where M is the total number of triplets, $\hat{p}_k$ is the k-th gold role label. We summarize them into total loss:

$$\mathcal{L} = \mathcal{L}_{pointer} + \mathcal{L}_{role} + \frac{\lambda}{2}||\Theta||^2\,, \tag{15}$$

where λ is a regularization for the ℓ_2 norm term, Θ is the overall parameters.

5 Experiment and Evaluation

Setup

We train and evaluate all models on two SRL benchmarks, including CoNLL09 (English), and Universal Proposition Bank (eight languages). We employ the official training, development and test sets in each dataset. The gold-standard syntactic features (POS tags and dependency trees) are also offered. In terms of hyper-parameters, since BERT is used, the size of word representations is 768.[2] The size of POS tag embeddings is 50. We use a 3-layer LA-GCN with 350 hidden units, and the output size of LSTM decoder is 300. We adopt the Adam optimizer with an initial learning rate 2e-5, mini-batch size 16 and regularization weight 0.12.

We consider two SRL setups: unified and pipeline SRL. For the unified SRL, we make comparisons with: He et al. (2018), Cai et al. (2018), Li et al. (2019), all of which is graph-based neural models. We equip all the baselines with a label-parameter GCN encoder (denoted as `LP-GCN`) for modeling syntactic dependency features (including arcs and labels). We also equip them with the vanilla GCN which does not encode dependency labels (denoted as `GCN*`). For the pipeline SRL, we use gold predicates for argument role labeling. We use precision (P), recall (R) and F1 scores for measuring the argument role labeling (**Arg**). For predicate disambiguation (**Prd**), only F1 is presented. All models are trained and evaluated 5 times and the averaged values are reported.

[2]https://github.com/google-research/bert, base-cased version.

	EN	DE	FR	IT	ES	PT	FI	ZH	*Avg.*
He et al. (2018)	87.0	80.0	89.6	78.5	79.4	80.7	78.2	71.6	80.6
Cai et al. (2018)	88.3	80.9	90.6	79.8	80.1	81.8	78.1	74.8	81.9
Li et al. (2019)	90.6	82.2	92.0	**81.3**	82.6	**85.1**	80.7	75.1	83.6
Ours	**91.5**	**84.5**	**93.8**	80.7	**84.0**	85.0	**82.1**	**76.8**	**84.9**

Table 2: F1s of argument role labeling for unified SRL on UPB. All models use BERT and the `LA-GCN` encoder.

	P	R	F1
Pipeline Model			
Zhao et al. (2009)	-	-	85.4*
Björkelund et al. (2010)	87.1*	84.5*	85.8*
FitzGerald et al. (2015)	-	-	86.7*
Roth and Lapata (2016)	88.1*	85.3*	86.7*
Marcheggiani and Titov (2017)	89.1*	86.8*	88.0*
Unified Model			
He et al. (2018)	89.8	89.6	89.7
Cai et al. (2018)	89.9	90.2	90.0
Li et al. (2019) +`LA-GCN` +`BERT`	92.3	92.6	92.4
Ours	**92.8**	**92.9**	**92.8**

Table 3: Argument role labeling with gold predicates on CoNLL09. Values with * are from prior papers.

	Arg	Prd
Ours	**93.2**	**97.4**
w/o POS	92.6	97.0
w/o BERT	90.8	95.7
w/o `LA-GCN`	91.2	95.9
w/o dependent label (`GCN*`)	91.9	96.2
`LP-GCN`	92.7	97.1
w/o Syntactic-aware Pointer	92.2	96.9
w/o HI Attention (Eq. 8)	92.0	96.5
base pointer (Eq. 1)	91.7	96.1
left-to-right parsing scheme	92.5	97.2
w/o residual connection (Eq. 6)	91.9	96.4

Table 4: Ablation results (F1) on CoNLL09.

Result and Analysis

Results on CoNLL09 English In Table 1, we have the following observations: First, all the models with the BERT enhanced representations are much stronger than those without BERT. This is consistent with prior BERT-based studies (He, Li, and Zhao 2019; Fei, Zhang, and Ji 2020). Second, the methods when modeling both syntactic dependent labels and arcs (i.e., `LP-GCN` and `LA-GCN`) consistently outperform those models using only dependent arcs (i.e., `GCN*`). Meanwhile, our proposed `LA-GCN` encoder presents quite improved results, compared with the `LP-GCN` encoder.

Most importantly, our framework significantly outperforms all the baselines on two test sets for both argument role labeling and predicate disambiguation. For example, by equipped with `LP-GCN` under no BERT representations, ours wins 0.3%(90.2-89.9) and 0.5%(79.7-79.2) F1 gaps than Li et al. (2019) for 'Arg' on *WSJ* and *Brown* respectively. With the `LA-GCN`, our model can further increase the wining gaps over baselines. The improvements largely lie in that the proposed syntactic-aware pointer mechanism is available when equipped with `LA-GCN` encoder, which additionally brings the enhancements. Also with our proposed `LA-GCN` plus BERT, the advantages can still be retained.

Results on UPB Table 2 shows the comparisons between different models on UPB. Above all, our pointer network wins the best overall results against all the graph-based baselines, with an average F1 score (84.9%). The improvements from our model demonstrate its effectiveness on unified SRL over all strong baselines.

Argument Role Labeling with Gold Predicates In Table 3, we show the performances for pipeline SRL based on CoNLL09 (*WSJ*) data. Compared with the traditional pipeline baselines, the unified systems can perform better. More importantly, with the help of `LA-GCN` encoder and BERT, our framework outperforms two best baselines, i.e., 92.8% F1, maintaining the advances on the standalone argument role labeling.

Ablation Study In Table 4, we list the influences of different proposed mechanisms on our model, based on the development set of CoNLL09 (*WSJ*). For the input features, the BERT representations are of the most prominent helpfulness. The dependency features (both arcs and labels) also show significant impacts, from the remove of the `LA-GCN`. Looking into the dependency features, either only decoding the dependent arcs (via `GCN*`) or replacing our `LA-GCN` with `LP-GCN`, the performances will drop.

Besides, both the syntactic-aware pointer mechanism and the high-order interacted attention mechanism plays important roles for the resulting framework. Especially when both two mechanisms are unavailable (degraded into the *base pointer*), the performance decreases crucially. Finally, the results fall if replacing the close-first parsing scheme with a vanilla left-to-right direction. Also canceling the residual connection of BERT representation results in substantial drops, as the utilities from rich BERT contexts are hurt.

Improvement for Argument Roles In Table 5, we present the results of high-frequency arguments on the UPB English test set. We compare with the best baseline Li et al. (2019) under different GCN encoders. We note that when our model takes `LP-GCN` encoder, the syntactic-aware pointer mechanism is unusable. Two models both use BERT representations for fair comparisons.

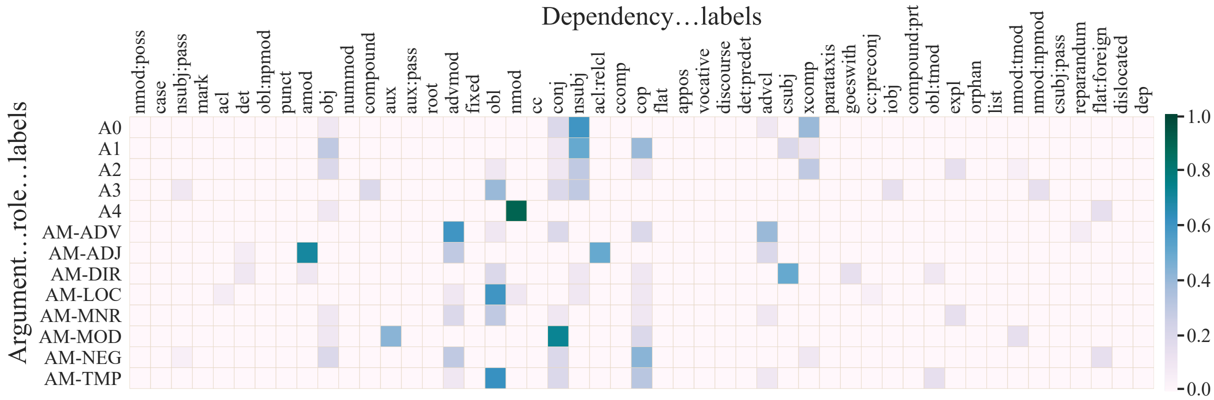

Figure 5: Correlations between dependency labels (Universal Dependency Treebank v1.4) and argument roles on UPB English.

	Li et al. (2019)		Ours	
	`GCN`*	`LP-GCN`	`LP-GCN`	`LA-GCN`
`A0`	85.6	86.3	87.2	**88.3**
`A1`	87.4	88.7	89.0	**89.8**
`A2`	82.9	83.2	84.1	**85.0**
`A3`	21.8	32.0	34.1	**41.2**
`A4`	50.0	53.4	52.3	**60.3**
`AM-ADV`	55.3	56.7	57.8	**58.2**
`AM-ADJ`	70.2	71.6	72.4	**75.4**
`AM-DIR`	34.7	36.0	38.8	**40.3**
`AM-DIS`	67.9	69.2	67.5	**70.1**
`AM-LOC`	49.6	50.4	50.0	**52.0**
`AM-MNR`	51.2	53.3	57.4	**62.4**
`AM-MOD`	95.1	96.3	96.8	**97.3**
`AM-NEG`	89.4	92.0	93.1	**96.0**
`AM-PRR`	89.0	90.5	91.6	**94.3**
`AM-TMP`	76.0	77.6	79.3	**82.0**
`C-A1`	56.4	61.9	63.1	**64.2**
`R-A0`	80.6	82.5	83.2	**84.5**
`R-A1`	60.5	62.2	67.3	**71.9**
Weighted Avg.	69.9	71.8	72.5	**73.3**

Table 5: Improvement for argument roles on UPB English.

First of all, when modeling the dependent type information, all argument roles receive improvements, which can be learnt by the Li et al. (2019) under `GCN`* and `LP-GCN`. Next, using the same `LP-GCN` encoder, our pointer network achieves the overall better scores (i.e., 72.5% weighted average F1) than Li et al. (2019). Most prominently, our full framework (with `LA-GCN`) can gain the improvements from all the role labels. Especially we find that the improvements for those less occurred arguments are more significant, such as `A3`, `A4`, `AM-ADJ` and `AM-MNR`. We give the credit to the stronger capability of our pointer network on the collaboratively learning between syntactic dependency structure and semantic role labeling. We also thank to the `LA-GCN` encoder for unifiedly modeling the dependent arcs and labels, where the yielded syntax connecting-strength distribution combined with the syntactic-aware pointer mechanism further contributes to the process.

Correlations between Dependency Labels and Semantic Roles Here we investigate the correlations between dependency labels and semantic roles, which are discovered by our framework. Technically, when a *predicate-argument* pair receives the argument role label (by Eq. 10), we first collect the syntactic weights (i.e., $\beta_{i,j}^{(L)}$) of each dependency labels concerning the predicate and argument tokens ($\boldsymbol{e}^p$ and $\boldsymbol{e}^a$ in Eq. 11), respectively. For each argument role, we then normalize these weights over each dependency label into distribution. We visualize the correlations between semantic roles and dependency labels in Figure 5.

We can learn that our framework has captured the underlying inter-dependency between the syntactic structure and the SRL structure from the diversified visualizations. Specifically, some interesting patterns can be observed. First, for each semantic role, only a small subset of dependent labels will contribute. For example, the `A1` role majorly relates to these *obj*, *nsubj*, *cop* and *csubj* dependent labels, while the `A4` role almost depends on the *nmod* label only. Such finding may provide a crucial foundation for the future works on the direction of unsupervised semantic role label that rely on the syntactic structures. Besides, we find that the framework tends to master such correlations accurately. This explains why our model can still achieve prominent improvement than the baseline model on those minority role labels, such as `A4`, `AM-TMP`, etc.

6 Conclusion

We proposed a unified SRL framework based on the pointer network. We further proposed a label-aware GCN (LA-GCN) encoder for modeling both syntactic dependent arcs and labels in a unified manner. Besides, a high-order interacted attention mechanism was introduced for leveraging prior recognized triplets into the current decision. Lastly, a syntactic-aware pointer was proposed which fully incorporated the syntactic dependent information yielded from LA-GCN. Our framework significantly outperforms existing graph-based SRL models on CoNLL09 and Universal Proposition Bank. Further analysis demonstrates that our model effectively captures the correlations between dependency labels and semantic roles.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61772378), the National Key Research and Development Program of China (No. 2017YFC1200500), the Research Foundation of Ministry of Education of China (No. 18JZD015).

References

Akbik, A.; Chiticariu, L.; Danilevsky, M.; Li, Y.; Vaithyanathan, S.; and Zhu, H. 2015. Generating high quality proposition banks for multilingual semantic role labeling. In *Proceedings of the ACL*, 397–407.

Akbik, A.; and Li, Y. 2016. Polyglot: Multilingual semantic role labeling with unified labels. In *Proceedings of the ACL*, 1–6.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the ICLR*.

Björkelund, A.; Bohnet, B.; Hafdell, L.; and Nugues, P. 2010. A High-Performance Syntactic and Semantic Dependency Parser. In *Proceedings of the Coling*, 33–36.

Cai, J.; He, S.; Li, Z.; and Zhao, H. 2018. A Full End-to-End Semantic Role Labeler, Syntactic-agnostic Over Syntactic-aware? In *Proceedings of the CoLing*, 2753–2765.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the NAACL*, 4171–4186.

Dozat, T.; and Manning, C. D. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *Proceedings of the ICLR*.

Fei, H.; Ren, Y.; and Ji, D. 2020. High-order Refining for End-to-end Chinese Semantic Role Labeling. In *Proceedings of the AACL*, 100–105.

Fei, H.; Zhang, M.; and Ji, D. 2020. Cross-Lingual Semantic Role Labeling with High-Quality Translated Training Corpus. In *Proceedings of the ACL*, 7014–7026.

Fei, H.; Zhang, M.; Li, F.; and Ji, D. 2020. Cross-lingual Semantic Role Labeling with Model Transfer. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28: 2427–2437.

Fernández-González, D.; and Gómez-Rodríguez, C. 2020. Transition-based Semantic Dependency Parsing with Pointer Networks. In *Proceedings of the ACL*, 7035–7046.

FitzGerald, N.; Täckström, O.; Ganchev, K.; and Das, D. 2015. Semantic Role Labeling with Neural Network Factors. In *Proceedings of the EMNLP*, 960–970.

Gildea, D.; and Jurafsky, D. 2000. Automatic Labeling of Semantic Roles. In *Proceedings of the ACL*, 512–520.

Hajič, J.; Ciaramita, M.; Johansson, R.; Kawahara, D.; Martí, M. A.; Màrquez, L.; Meyers, A.; Nivre, J.; Padó, S.; Štěpánek, J.; Straňák, P.; Surdeanu, M.; Xue, N.; and Zhang, Y. 2009. The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In *Proceedings of the CoNLL*, 1–18.

He, L.; Lee, K.; Levy, O.; and Zettlemoyer, L. 2018. Jointly Predicting Predicates and Arguments in Neural Semantic Role Labeling. In *Proceedings of the ACL*, 364–369.

He, S.; Li, Z.; and Zhao, H. 2019. Syntax-aware Multilingual Semantic Role Labeling. In *Proceedings of the EMNLP*, 5353–5362.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8): 1735–1780.

Jin, G.; Kawahara, D.; and Kurohashi, S. 2015. Chinese Semantic Role Labeling using High-quality Syntactic Knowledge. In *Proceedings of the IJCNLP*, 120–127.

Kasai, J.; Friedman, D.; Frank, R.; Radev, D.; and Rambow, O. 2019. Syntax-aware Neural Semantic Role Labeling with Supertags. In *Proceedings of the NAACL*, 701–709.

Lei, T.; Zhang, Y.; Màrquez, L.; Moschitti, A.; and Barzilay, R. 2015. High-Order Low-Rank Tensors for Semantic Role Labeling. In *Proceedings of the NAACL*, 1150–1160.

Li, J.; Ye, D.; and Shang, S. 2019. Adversarial Transfer for Named Entity Boundary Detection with Pointer Networks. In *Proceedings of the IJCAI*, 5053–5059.

Li, Z.; He, S.; Zhao, H.; Zhang, Y.; Zhang, Z.; Zhou, X.; and Zhou, X. 2019. Dependency or Span, End-to-End Uniform Semantic Role Labeling. In *Proceedings of the AAAI*, 6730–6737.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the EMNLP*, 1412–1421.

Ma, X.; Hu, Z.; Liu, J.; Peng, N.; Neubig, G.; and Hovy, E. 2018. Stack-Pointer Networks for Dependency Parsing. In *Proceedings of the ACL*, 1403–1414.

Marcheggiani, D.; Frolov, A.; and Titov, I. 2017. A Simple and Accurate Syntax-Agnostic Neural Model for Dependency-based Semantic Role Labeling. In *Proceedings of the CoNLL*, 411–420.

Marcheggiani, D.; and Titov, I. 2017. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. In *Proceedings of the EMNLP*, 1506–1515.

Pradhan, S.; Ward, W.; Hacioglu, K.; Martin, J.; and Jurafsky, D. 2005. Semantic Role Labeling Using Different Syntactic Views. In *Proceedings of the ACL*, 581–588.

Punyakanok, V.; Roth, D.; and Yih, W. 2008. The Importance of Syntactic Parsing and Inference in Semantic Role Labeling. *Computational Linguistics* 34(2): 257–287.

Roth, M.; and Lapata, M. 2016. Neural Semantic Role Labeling with Dependency Path Embeddings. In *Proceedings of the ACL*, 1192–1202.

Scheible, C. 2010. An Evaluation of Predicate Argument Clustering using Pseudo-Disambiguation. In *Proceedings of the LREC*.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the ACL*, 1073–1083.

Strubell, E.; Verga, P.; Andor, D.; Weiss, D.; and McCallum, A. 2018. Linguistically-Informed Self-Attention for Semantic Role Labeling. In *Proceedings of the EMNLP*, 5027–5038.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the NeurIPS*, 3104–3112.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Proceedings of the NeurIPS*, 5998–6008.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer Networks. In *Proceedings of the NeurIPS*, 2692–2700.

Williams, R. J.; and Zipser, D. 1989. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation* 1(2): 270–280.

Xia, Q.; Li, Z.; Zhang, M.; Zhang, M.; Fu, G.; Wang, R.; and Si, L. 2019. Syntax-Aware Neural Semantic Role Labeling. In *Proceedings of the AAAI*, 7305–7313.

Zhang, Y.; Wang, R.; and Si, L. 2019. Syntax-Enhanced Self-Attention-Based Semantic Role Labeling. In *Proceedings of the EMNLP*, 616–626.

Zhao, H.; Chen, W.; Kazama, J.; Uchimoto, K.; and Torisawa, K. 2009. Multilingual Dependency Learning: Exploiting Rich Features for Tagging Syntactic and Semantic Dependencies. In *Proceedings of the CoNLL*, 61–66.