Dependency or Span, End-to-End Uniform Semantic Role Labeling

Zuchao Li,[1,2,*] **Shexia He,**[1,2,*] **Hai Zhao,**[1,2,†] **Yiqing Zhang,**[1,2] **Zhuosheng Zhang,**[1,2]
Xi Zhou,[3] **Xiang Zhou**[3]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
[3]CloudWalk Technology, Shanghai, China
{charlee,heshexia}@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn,
{zhangyiqing,zhangzs}@sjtu.edu.cn, {zhouxi,zhouxiang}@cloudwalk.cn

Abstract

Semantic role labeling (SRL) aims to discover the predicate-argument structure of a sentence. End-to-end SRL without syntactic input has received great attention. However, most of them focus on either span-based or dependency-based semantic representation form and only show specific model optimization respectively. Meanwhile, handling these two SRL tasks uniformly was less successful. This paper presents an end-to-end model for both dependency and span SRL with a unified argument representation to deal with two different types of argument annotations in a uniform fashion. Furthermore, we jointly predict all predicates and arguments, especially including long-term ignored predicate identification subtask. Our single model achieves new state-of-the-art results on both span (CoNLL 2005, 2012) and dependency (CoNLL 2008, 2009) SRL benchmarks.

Introduction

The purpose of semantic role labeling (SRL) is to derive the meaning representation for a sentence, which is beneficial to a wide range of natural language processing (NLP) tasks (Wang et al. 2016; Zhang et al. 2018). SRL can be formed as four subtasks, including predicate detection, predicate disambiguation, argument identification and argument classification. For argument annotation, there are two formulizations. One is based on text spans, namely span-based SRL. The other is dependency-based SRL, which annotates the syntactic head of argument rather than entire argument span. Figure 1 shows example annotations.

Great progress has been made in syntactic parsing (Dozat and Manning 2017; Li et al. 2018a; 2018c). Most traditional SRL methods rely heavily on syntactic features. To alleviate the inconvenience, recent works (Zhou and Xu 2015; Marcheggiani, Frolov, and Titov 2017; He et al. 2017; Tan et al. 2018; He et al. 2018a; 2018b; Cai et al. 2018) propose end-to-end models for SRL, putting syntax aside and still achieving favorable results. However, these systems focus on either span or dependency SRL, which motivates us to explore a uniform approach.

A0	*v*	A1		A2		AM-TMP	
Marry	*borrowed*	a	book	from	John	last	week
A0	*borrow.01*		A1	A2			AM-TMP

Figure 1: Examples of annotations in span (above) and dependency (below) SRL.

Both span and dependency are effective formal representations for semantics, though for a long time it has been kept unknown which form, span or dependency, would be better for the convenience and effectiveness of semantic machine learning and later applications. Furthermore, researchers are interested in two forms of SRL models that may benefit from each other rather than their separated development. This topic has been roughly discussed in (Johansson and Nugues 2008a), who concluded that the (best) dependency SRL system at then clearly outperformed the span-based (best) system through gold syntactic structure transformation. However, Johansson and Nugues (2008a) like all other traditional SRL models themselves had to adopt rich syntactic features, and their comparison was done between two systems in quite different building styles. Instead, this work will develop full syntax-agnostic SRL systems with the same fashion for both span and dependency representation, so that we can revisit this issue under a more solid empirical basis.

In addition, most efforts focus on argument identification and classification since span and dependency SRL corpora have already marked predicate positions. Although no predicate identification is needed, it is not available in many downstream applications. Therefore, predicate identification should be carefully handled in a complete practical SRL system. To address this problem, He et al. (2018a) proposed an end-to-end approach for jointly predicting predicates and arguments for span SRL. Likewise, Cai et al. (2018) introduced an end-to-end model to naturally cover all predicate/argument identification and classification subtasks for dependency SRL.

To jointly predict predicates and arguments, we present an end-to-end framework for both span and dependency SRL.

* These authors made equal contribution.† Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100), National Natural Science Foundation of China (No. 61672343 and No. 61733011), Key Project of National Society Science Foundation of China (No. 15-ZDA041), The Art and Science Interdisciplinary Funds of Shanghai Jiao Tong University (No. 14JCRZ04).

Span (CoNLL 2005)						Dependency (CoNLL 2009)					
Time	System	SA	ST	Method	$\mathbf{F}_1$	**Time**	System	SA	ST	Method	$\boldsymbol{F}_1$
2008	Punyakanok et al.	+		ILP	**76.3**	**2009**	Zhao et al.	+		ME	***86.2***
2008	Toutanova et al.	+		DP	**79.7**	**2010**	Björkelund et al.	+		global	***86.9***
2015	**FitzGerald et al.**	+		structured	**79.4**			+		structured	***87.3***
2015	Zhou and Xu		+	deep BiLSTM	**82.8**						
						2016	Roth and Lapata	+		PathLSTM	***87.7***
2017	He et al.		+	highway BiLSTM	**83.1**	**2017**	Marcheggiani et al.		+	BiLSTM	***87.7***
						2017	Marcheggiani and Titov	+	+	GCNs	***88.0***
2018	Tan et al.		+	self-attention	**84.8**	**2018**	He et al. (b)	+	+	ELMo	***89.5***
2018	Strubell et al.	+		self-attention	**83.9**	**2018**	Cai et al.			biaffine	***89.6***
2018	He et al. (a)			ELMo	**87.4**	**2018**	Li et al. (a)	+	+	ELMo	***89.8***
2019	Li et al. (b) AAAI			ELMo+biaffine	**87.7**					ELMo+biaffine	***90.4***

Table 1: A chronicle of related work for span and dependency SRL. SA represents syntax-aware system (no + indicates syntax-agnostic system) and ST indicates sequence tagging model. F_1 is the result of single model on official test set.

Our model extends the span SRL model of He et al. (2018a), directly regarding all words in a sentence as possible predicates, considering all spans or words as potential arguments and learning distributions over possible predicates. However, we differ by (1) introducing unified argument representation to handle two different types of SRL tasks, and (2) employing biaffine scorer to make decisions for predicate-argument relationship.

The proposed models are evaluated on span SRL datasets: CoNLL 2005 and 2012 data, as well as the dependency SRL dataset of CoNLL 2008 and 2009 shared tasks. For span SRL, our single model outperforms the previous best results by 0.3% and 0.5% F_1-score on CoNLL 2005 and 2012 test sets respectively. For dependency SRL, we achieve new state-of-the-art of 85.3% F_1 and 90.4% F_1 on CoNLL 2008 and 2009 benchmarks respectively.

Background

SRL is pioneered by Gildea and Jurafsky (2002), which uses the PropBank conventions (Palmer, Gildea, and Kingsbury 2005). Conventionally, span SRL consists of two subtasks, argument identification and classification. The former identifies the arguments of a predicate, and the latter assigns them semantic role labels, namely, determining the relation between arguments and predicates. The PropBank defines a set of semantic roles to label arguments, falling into two categories: *core* and *non-core* roles. The core roles (A0-A5 and AA) indicate different semantics in predicate-argument structure, while the non-core roles are modifiers (AM-*adj*) where *adj* specifies the adjunct type, such as temporal (AM-TMP) and locative (AM-LOC) adjuncts. For example shown in Figure 1, A0 is a proto-agent, representing the *borrower*.

Slightly different from span SRL in argument annotation, dependency SRL labels the syntactic heads of arguments rather than phrasal arguments, which was popularized by CoNLL-2008 and CoNLL-2009 shared tasks[1] (Surdeanu et al. 2008; Hajič et al. 2009). Furthermore, when no predicate is given, two other indispensable subtasks of dependency SRL are predicate identification and disambiguation. One is to identify all predicates in a sentence, and the other is to determine the senses of predicates. As the example shown in Figure 1, *01* indicates the first sense from the PropBank sense repository for predicate *borrowed* in the sentence.

Related Work

The traditional approaches on SRL were mostly about designing hand-crafted feature templates and then employ linear classifiers such as (Pradhan et al. 2005; Punyakanok, Roth, and Yih 2008; Zhao et al. 2009). Even though neural models were introduced, early work still paid more attention on syntactic features. For example, FitzGerald et al. (2015) integrated syntactic information into neural networks with embedded lexicalized features, while Roth and Lapata (2016) embedded syntactic dependency paths between predicates and arguments. Similarly, Marcheggiani and Titov (2017) leveraged the graph convolutional network to encode syntax for dependency SRL. Recently, Strubell et al. (2018) presented a multi-task neural model to incorporate auxiliary syntactic information for SRL, Li et al. (2018b) adopted several kinds of syntactic encoder for syntax encoding while He et al. (2018b) used syntactic tree for argument pruning.

However, using syntax may be quite inconvenient sometimes, recent studies thus have attempted to build SRL systems without or with little syntactic guideline. Zhou and Xu (2015) proposed the first syntax-agnostic model for span SRL using LSTM sequence labeling, while He et al. (2017) further enhanced their model using highway bidirectional LSTMs with constrained decoding. Later, Tan et al. (2018) presented a deep attentional neural network for applying self-attention to span SRL task. Likewise for dependency SRL, Marcheggiani, Frolov, and Titov (2017) proposed a

[1]CoNLL-2008 is an English-only task, while CoNLL-2009 extends to a multilingual one. Their main difference is that predicates have been beforehand indicated for the latter. Or rather, CoNLL-2009 does not need predicate identification, but it is an indispensable subtask for CoNLL-2008.

syntax-agnostic model with effective word representation and obtained favorable results. Cai et al. (2018) built a full end-to-end model with biaffine attention and outperformed the previous state-of-the-art.

More recently, joint predicting both predicates and arguments has attracted extensive interest on account of the importance of predicate identification, including (He et al. 2017; Strubell et al. 2018; He et al. 2018a; Cai et al. 2018) and this work. In our preliminary experiments, we tried to integrate the self-attention into our model, but it does not provide any significant performance gain on span or dependency SRL, which is not consistent with the conclusion in (Tan et al. 2018) and lets us exclude it from this work.

Generally, the above work is summarized in Table 1. Considering motivation, our work is most closely related to the work of FitzGerald et al. (2015), which also tackles span and dependency SRL in a uniform fashion. The essential difference is that their model employs the syntactic features and takes pre-identified predicates as inputs, while our model puts syntax aside and jointly learns and predicts predicates and arguments.

Uniform End-to-End Model

Overview

Given a sentence $s = w_1, w_2, \ldots, w_n$, we attempt to predict a set of predicate-argument-relation tuples $\mathcal{Y} \in \mathcal{P} \times \mathcal{A} \times \mathcal{R}$, where $\mathcal{P} = \{w_1, w_2, ..., w_n\}$ is the set of all possible predicate tokens, $\mathcal{A} = \{(w_i, \ldots, w_j) | 1 \leq i \leq j \leq n\}$ includes all the candidate argument spans or dependencies[2], and $\mathcal{R}$ is the set of the semantic roles. To simplify the task, we introduce a null label ϵ to indicate no relation between arbitrary predicate-argument pair following He et al. (2018a). As shown in Figure 2, our uniform SRL model includes four main modules:

• token representation component to build token representation x_i from word w_i,

• a BiHLSTM encoder that directly takes sequential inputs,

• predicate and argument representation module to learn candidate representations,

• a biaffine scorer which takes the candidate representations as input and predicts semantic roles.

Token Representation

We follow the bi-directional LSTM-CNN architecture (Chiu and Nichols 2016), where convolutional neural networks (CNNs) encode characters inside a word w into character-level representation w_{char} then concatenated with its word-level w_{word} into context-independent representation. To further enhance the word representation, we leverage an external representation w_{elmo} from pretrained ELMo (Embeddings from Language Models) layers according to Peters et al. (2018). Eventually, the resulting token representation is concatenated as

$$x = [w_{char}, w_{word}, w_{elmo}].$$

[2]When i=j, it means span degrades to dependency.

Deep Encoder

The encoder in our model adopts the bidirectional LSTM with highway connections (BiHLSTM) to contextualize the representation into task-specific representation: $x_i^c \in X^c$: $X^c = \textit{BiHLSTM}(\{x_i\})$, where the gated highway connections is used to alleviate the vanishing gradient problem when training very deep BiLSTMs.

Predicate and Argument Representation

We employ contextualized representations for all candidate arguments and predicates. As referred in (Dozat and Manning 2017), applying a multi-layer perceptron (MLP) to the recurrent output states before the classifier has the advantage of stripping away irrelevant information for the current decision. Therefore, to distinguish the currently considered predicate from its candidate arguments in SRL context, we add an MLP layer to contextualized representations for argument g^a and predicate g^p candidates specific representations respectively with ReLU (Nair and Hinton 2010) as its activation function:

$$g^a = \textit{ReLU}(\mathbf{MLP}_a(X^c))$$
$$g^p = \textit{ReLU}(\mathbf{MLP}_p(X^c))$$

To perform uniform SRL, we introduce unified argument representation. For dependency SRL, we assume single word argument span by limiting the length of candidate argument to be 1, so our model uses the g^a as the final argument representation g_f^a directly. While for span SRL, we utilize the approach of span representation from Lee et al. (2017). Each candidate span representation g_f^a is built by

$$g_f^a = [g^a_{\textit{START}}, g^a_{\textit{END}}, h_\lambda, size(\lambda)],$$

where g^a_{START} and g^a_{END} are boundary representations, λ indicates a span, $size(\lambda)$ is a feature vector encoding the size of span, and h_λ is the specific notion of headedness which is learned by attention mechanism (Bahdanau, Cho, and Bengio 2014) over words in each span (where t is the position inside span) as follows :

$$\mu_t^a = \mathbf{w}_{attn} \cdot \mathbf{MLP}_{attn}(g_t^a)$$
$$\nu_t = \frac{\exp(\mu_t^a)}{\sum_{k=\textit{START}}^{\textit{END}} \exp(\mu_k^a)}$$
$$h_\lambda = \sum_{t=\textit{START}}^{\textit{END}} \nu_t \cdot g_t^a$$

Scorers

For predicate and arguments, we introduce two unary scores on their candidates:

$$\phi_p = \mathbf{w}_p \mathbf{MLP}_p^s(g^p),$$
$$\phi_a = \mathbf{w}_a \mathbf{MLP}_a^s(g_f^a).$$

For semantic role, we adopt a relation scorer with biaffine attention (Dozat and Manning 2017):

$$\Phi_r(p, a) = \textit{Biaffine}(g^p, g_f^a)$$
$$= \{g_t^p\}^T \mathbf{W}_1 g_f^a \quad (1)$$
$$+ \mathbf{W}_2^T (g_t^p \oplus g_f^a) + \mathbf{b} \quad (2)$$

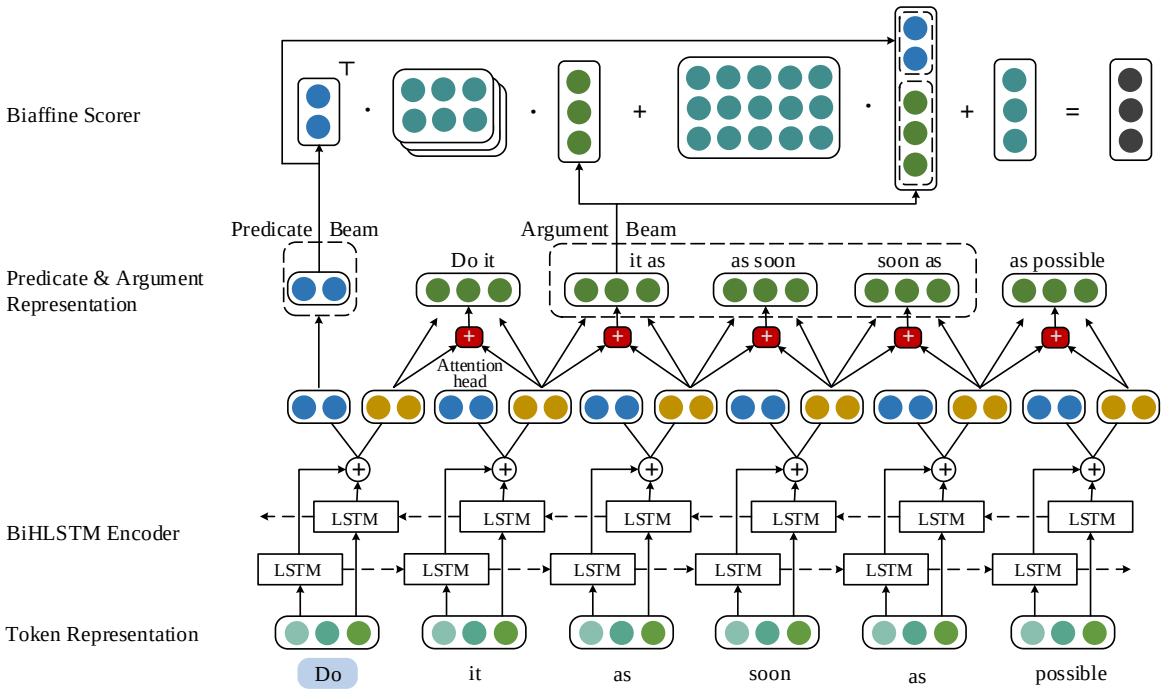

Figure 2: The framework of our end-to-end model for uniform SRL.

where $\mathbf{W}_1$ and $\mathbf{W}_2$ respectively denote the weight matrix of the bi-linear and the linear terms and $\mathbf{b}$ is the bias item.

The biaffine scorer differs from feed-forward networks scorer in bilinear transformation. Since SRL can be regarded as a classification task, the distribution of classes is uneven and the problem comes worse after the null labels are introduced. The output layer of the model normally includes a bias term designed to capture the prior probability of each class, with the rest of the model focusing on learning the likelihood of every classes occurring in data. The biaffine attention as Dozat and Manning (2017) in our model directly assigns a score for each specific semantic role and would be helpful for semantic role prediction. Actually, (He et al., 2018a) used a scorer as Equation (2), which is only a part of our scorer including both Equations (1) and (2). Therefore, our scorer would be more informative than previous models such as (He et al. 2018a).

Training Objective

The model is trained to optimize the probability $P_\theta(\hat{y}|s)$ of the predicate-argument-relation tuples $\hat{y}_{(p,a,r)} \in \mathcal{Y}$ given the sentence s, which can be factorized as:

$$P_\theta(y|s) = \prod_{p\in\mathcal{P},a\in\mathcal{A},r\in\mathcal{R}} P_\theta(y_{(p,a,r)}|s)$$
$$= \prod_{p\in\mathcal{P},a\in\mathcal{A},r\in\mathcal{R}} \frac{\phi(p,a,r)}{\sum_{\hat{r}\in\mathcal{R}}\phi(p,a,\hat{r})}$$

where θ represents the model parameters, and $\phi(p,a,r) = \phi_p + \phi_a + \Phi_r(p,a)$, is the score for the predicate-argument-relation tuple, including predicate score ϕ_p, argument score ϕ_a and relation score $\Phi_r(p,a)$.

Our model adopts a biaffine scorer for semantic role label prediction, which is implemented as cross-entropy loss. Moreover, our model is trained to minimize the negative likehood of the golden structure y: $\mathcal{J}(s) = -\log P_\theta(y|s)$. The score of null labels are enforced into $\phi(p,a,\epsilon) = 0$. For predicates and arguments prediction, we train separated scorers (ϕ_p and ϕ_a) in parallel fed to the biaffine scorer for predicate and argument predication respectively, which helps to reduce the chance of error propagation.

Candidates Pruning

The number of candidate arguments for a sentence of length n is $O(n^2)$ for span SRL, and $O(n)$ for dependency. As the model deals with $O(n)$ possible predicates, the computational complexity is $O(n^3 \cdot |\mathcal{R}|)$ for span, $O(n^2 \cdot |\mathcal{R}|)$ for dependency, which is too computationally expensive.

To address this issue, we attempt to prune candidates using two beams for storing the candidate arguments and predicates with size $\beta_p n$ and $\beta_a n$ inspired by He et al. (2018a), where β_p and β_a are two manually setting thresholds. First, the predicate and argument candidates are ranked according to their predicted score (ϕ_p and ϕ_a) respectively, and then we reduce the predicate and argument candidates with defined beams. Finally, we take the candidates from the beams to participate the label prediction. Such pruning will reduce the overall number of candidate tuples to $O(n^2 \cdot |\mathcal{R}|)$ for both types of tasks. Furthermore, for span SRL, we set the maximum length of candidate arguments to $\mathcal{L}$, which may decrease the number of candidate arguments to $O(n)$.

End-to-End	CoNLL-2005 WSJ			CoNLL-2005 Brown			CoNLL-2012 (OntoNotes)		
	P	R	F_1	P	R	F_1	P	R	F_1
He et al. (2017) (Single)	80.2	82.3	81.2	67.6	69.6	68.5	78.6	75.1	76.8
Strubell et al. (2018)	83.7	83.7	83.7	72.6	69.7	71.1	80.7	79.1	79.9
He et al. (2018a)	84.8	87.2	86.0	73.9	**78.4**	76.1	81.9	**84.0**	82.9
Ours (Single)	**85.2**	**87.5**	**86.3**	**74.7**	78.1	**76.4**	**84.9**	81.4	**83.1**
He et al. (2017) (Ensemble)	82.0	83.4	82.7	69.7	70.5	70.1	80.2	76.6	78.4

Table 2: End-to-end span SRL results on CoNLL-2005 and CoNLL-2012 data, compared with previous systems in terms of precision (P), recall (R), F_1-score. The CoNLL-2005 contains two test sets: WSJ (in-domain) and Brown (out-of-domain).

Systems	WSJ			Brown		
	P	R	F_1	P	R	F_1
J & N (2008b)	–	–	81.75	–	–	69.06
Zhao et al. (2009)	–	–	82.1	–	–	–
Zhao et al. (2013)	–	–	82.5	–	–	–
He et al. (2018b)	83.9	82.7	83.3	–	–	–
Cai et al.(2018)	**84.7**	85.2	85.0	–	–	72.5
Ours	84.5	**86.1**	**85.3**	74.6	73.8	**74.2**

Table 3: Dependency SRL results on CoNLL-2008 test sets.

SRL Constraints

According to PropBank semantic convention, predicate-argument structure has to follow a few of global constraints (Punyakanok, Roth, and Yih 2008; He et al. 2017), we thus incorporate constraints on the output structure with a dynamic programing decoder during inference. These constraints are described as follows:

• Unique core roles (U): Each core role (A0-A5, AA) should appear at most once for each predicate.

• Continuation roles (C): A continuation role C-X can exist only when its base role X is realized before it.

• Reference roles (R): A reference role R-X can exist only when its base role X is realized (not necessarily before R-X).

• Non-overlapping (O): The semantic arguments for the same predicate do not overlap in span SRL.

As C and R constraints lead to worse performance in our models from our preliminary experiments, we only enforce U and O constraints on span SRL and U constraints on dependency SRL[3].

Experiments

Our models[4] are evaluated on two PropBank-style SRL tasks: span and dependency. For span SRL, we test model on the common span SRL datasets from CoNLL-2005 (Carreras and Màrquez 2005) and CoNLL-2012 (Pradhan et al. 2013) shared tasks. For dependency SRL, we experiment on CoNLL 2008 (Surdeanu et al. 2008) and 2009 (Hajič et al. 2009) benchmarks. As for the predicate disambiguation in dependency SRL task, we follow the previous work (Roth and Lapata 2016).

We consider two SRL setups: *end-to-end* and *pre-identified predicates*. For the former setup, our system jointly predicts all the predicates and their arguments in one shot, which turns into CoNLL-2008 setting for dependency SRL. In order to compare with previous models, we also report results with *pre-identified predicates*, where predicates have been beforehand identified in corpora. Therefore, the experimental results fall into two categories: end-to-end results and results with pre-identified predicates.

Datasets

CoNLL 2005 and 2012 The CoNLL-2005 shared task focused on verbal predicates only for English. The CoNLL-2005 dataset takes section 2-21 of Wall Street Journal (WSJ) data as training set, and section 24 as development set. The test set consists of section 23 of WSJ for in-domain evaluation together with 3 sections from Brown corpus for out-of-domain evaluation. The larger CoNLL-2012 dataset is extracted from OntoNotes v5.0 corpus, which contains both verbal and nominal predicates.

CoNLL 2008 and 2009 CoNLL-2008 and the English part of CoNLL-2009 shared tasks use the same English corpus, which merges two treebanks, PropBank and NomBank. NomBank is a complement to PropBank with similar semantic convention for nominal predicate-argument structure annotation. Besides, the training, development and test splits of English data are identical to that of CoNLL-2005.

Setup

Hyperparameters In our experiments, the word embeddings are 300-dimensional GloVe vectors (Pennington, Socher, and Manning 2014). The character representations with dimension 8 randomly initialized. In the character CNN, the convolutions have window sizes of 3, 4, and 5, each consisting of 50 filters. Moreover, we use 3 stacked bidirectional LSTMs with 200 dimensional hidden states. The outputs of BiLSTM employs two 300-dimensional MLP layers with the ReLU as activation function. Besides, we use two 150-dimensional hidden MLP layers with ReLU to score predicates and arguments respectively. For candidates pruning, we follow the settings of He et al. (2018a), modeling spans up to length $\mathcal{L} = 30$ for span SRL and $\mathcal{L} = 1$ for

[3]O constraint will be automatically satisfied for dependency, as it may be regarded as length 1 sized span.

[4]Our code is available here: https://github.com/bcmi220/unisrl.

	System	CoNLL-2005 WSJ			CoNLL-2005 Brown			CoNLL-2012 (OntoNotes)		
		P	R	F_1	P	R	F_1	P	R	F_1
Single	Pradhan et al. (2013) (Revised)	–	–	–	–	–	–	78.5	76.6	77.5
	Zhou and Xu (2015)	82.9	82.8	82.8	70.7	68.2	69.4	–	–	81.3
	He et al. (2017)	83.1	83.0	83.1	72.9	71.4	72.1	81.7	81.6	81.7
	Tan et al. (2018)	84.5	85.2	84.8	73.5	74.6	74.1	81.9	83.6	82.7
	Peters et al. (2018)	–	–	–	–	–	–	–	–	84.6
	Strubell et al. (2018)	83.9	83.9	83.9	73.3	71.8	72.5	–	–	–
	He et al. (2018a)	–	–	87.4	–	–	80.4	–	–	85.5
	Ours	**87.9**	**87.5**	**87.7**	**80.6**	**80.4**	**80.5**	**85.7**	**86.3**	**86.0**
Ensemble	Punyakanok et al. (2008)	82.3	76.8	79.4	73.4	62.9	67.8	–	–	–
	Toutanova et al. (2008)	81.9	78.8	80.3	–	–	68.8	–	–	–
	FitzGerald et al. (2015)	82.5	78.2	80.3	74.5	70.0	72.2	81.2	79.0	80.1
	He et al. (2017)	85.0	84.3	84.6	74.9	72.4	73.6	83.5	83.3	83.4
	Tan et al. (2018)	85.9	86.3	86.1	74.6	75.0	74.8	83.3	84.5	83.9

Table 4: Span SRL results with pre-identified predicates on CoNLL-2005 and CoNLL-2012 test sets.

dependency SRL, using $\beta_p = 0.4$ for pruning predicates and $\beta_a = 0.8$ for pruning arguments.

Training Details During training, we use the categorical cross-entropy as objective, with Adam optimizer (Kingma and Ba 2015) initial learning rate 0.001. We apply 0.5 dropout to the word embeddings and character CNN outputs and 0.2 dropout to all hidden layers and feature embeddings. In the LSTMs, we employ variational dropout masks that are shared across timesteps (Gal and Ghahramani 2016), with 0.4 dropout rate. All models are trained for up to 600 epochs with batch size 40 on a single NVIDIA GeForce GTX 1080Ti GPU, which occupies 8 GB graphic memory and takes 12 to 36 hours.

End-to-end Results

We present all results using the official evaluation script from the CoNLL-2005 and CoNLL-2009 shared tasks, and compare our model with previous state-of-the-art models.

Span SRL Table 2 shows results on CoNLL-2005 in-domain (WSJ) and out-of-domain (Brown) test sets, as well as the CoNLL-2012 test set (OntoNotes). The upper part of table presents results from single models. Our model outperforms the previous models with absolute improvements in F_1-score of 0.3% on CoNLL-2005 benchmark. Besides, our single model performs even much better than all previous ensemble systems.

Dependency SRL Table 3 presents the results on CoNLL-2008. J & N (2008b) (Johansson and Nugues 2008b) was the highest ranked system in CoNLL-2008 shared task. We obtain comparable results with the recent state-of-the-art method (Cai et al. 2018), and our model surpasses the model (He et al. 2018b) by 2% in F_1-score.

Results with Pre-identified Predicates

To compare with to previous systems with pre-identified predicates, we report results from our models as well.

Span SRL Table 4 shows that our model outperforms all published systems, even the ensemble model (Tan et al. 2018), achieving the best results of 87.7%, 80.5% and 86.0% in F_1-score respectively.

Dependency SRL Table 5 compares the results of dependency SRL on CoNLL-2009 English data. Our single model gives a new state-of-the-art result of 90.4% F_1 on WSJ. For Brown data, the proposed syntax-agnostic model yields a performance gain of 1.7% F_1 over the syntax-aware model (Li et al. 2018b).

Ablation

To investigate the contributions of ELMo representations and biaffine scorer in our end-to-end model, we conduct a series of ablation studies on the CoNLL-2005 and CoNLL-2008 WSJ test sets, unless otherwise stated.

Table 6 compares F_1 scores of He et al. (2018a) and our model without ELMo representations. We observe that effect of ELMo is somewhat surprising, where removal of the ELMo dramatically declines the performance by 3.3-3.5 F_1 on CoNLL-2005 WSJ. However, our model gives quite stable performance for dependency SRL regardless of whether ELMo is concatenated or not. The results indicate that ELMo is more beneficial to span SRL.

In order to better understand how the biaffine scorer influences our model performance, we train our model with different scoring functions. To ensure a fair comparison with the model (He et al. 2018a), we replace the biaffine scorer with their scoring functions implemented with feed-forward networks, and the results of removing biaffine scorer are also presented in Table 6. We can see 0.5% and 1.6% F_1 performance degradation on CoNLL 2005 and 2008 WSJ respectively. The comparison shows that the biaffine scorer is more effective for scoring the relations between predicates and arguments. Furthermore, these results show that biaffine attention mechanism is applicable to span SRL.

Dependency or Span?

It is very hard to say which style of semantic formal representation, dependency or span, would be more convenient

	System	CoNLL-2009 WSJ			CoNLL-2009 Brown		
		P	R	F_1	P	R	F_1
Single	Zhao et al. (2009)	–	–	86.2	–	–	74.6
	FitzGerald et al. (2015) (Struct.)	–	–	87.3	–	–	75.2
	Roth and Lapata (2016) (Global)	90.0	85.5	87.7	78.6	73.8	76.1
	Marcheggiani et al. (2017)	88.7	86.8	87.7	79.4	76.2	77.7
	Marcheggiani and Titov (2017)	89.1	86.8	88.0	78.5	75.9	77.2
	He et al. (2018b)	89.7	89.3	89.5	**81.9**	76.9	79.3
	Cai et al. (2018)	89.9	89.2	89.6	79.8	78.3	79.0
	Li et al. (2018b)	90.3	89.3	89.8	80.6	79.0	79.8
	Ours	89.6	**91.2**	**90.4**	81.7	**81.4**	**81.5**
Ensemble	FitzGerald et al. (2015)	–	–	87.7	–	–	75.5
	Roth and Lapata (2016)	90.3	85.7	87.9	79.7	73.6	76.5
	Marcheggiani and Titov (2017)	**90.5**	87.7	89.1	80.8	77.1	78.9

Table 5: Dependency SRL results with pre-identified predicates on CoNLL-2009 English benchmark.

System	CoNLL-2005	CoNLL-2008
Cai et al. (2018)	–	85.0
He et al. (2018a)	86.0	–
w/o ELMo	82.5	–
Ours	86.3	85.3
w/o ELMo	83.0	85.1
w/o biaffine scorer	85.8	83.7

Table 6: Effectiveness of ELMo representations and biaffine scorer on the CoNLL 2005 and 2008 WSJ sets.

for machine learning as they adopt incomparable evaluation metric. Recent researches (Peng et al. 2018) have proposed to learn semantic parsers from multiple datasets in Framenet style semantics, while our goal is to compare the quality of different models in the span and dependency SRL for Propbank style semantics. Following Johansson and Nugues (2008a), we choose to directly compare their performance in terms of dependency-style metric through a transformation way. Using the head-finding algorithm in (Johansson and Nugues 2008a) which used gold-standard syntax, we may determine a set of head nodes for each span. This process will output an upper bound performance measure about the span conversion due to the use of gold syntax.

We do not train new models for the conversion and the resulted comparison. Instead, we do the job on span-style CoNLL 2005 test set and dependency-style CoNLL 2009 test set (WSJ and Brown), considering these two test sets share the same text content. As the former only contains verbal predicate-argument structures, for the latter, we discard all nomial predicate-argument related results and predicate disambiguation results during performance statistics. Table 7 shows the comparison.

On a more strict setting basis, the results from our same model for span and dependency SRL verify the same conclusion of Johansson and Nugues (2008a), namely, dependency form is in a favor of machine learning effectiveness for SRL even compared to the conversion upper bound of span form.

		Dep F_1	Span-converted F_1	Δ F_1
WSJ	J & N	85.93	84.32	1.61
	Our system	90.41	89.20	1.21
WSJ+ Brown	J & N	84.29	83.45	0.84
	Our system	88.91	88.23	0.68

Table 7: Dependency vs. Span-converted Dependency on CoNLL 2005, 2009 test sets with dependency evaluation.

Conclusion

This paper presents an end-to-end neural model for both span and dependency SRL, which may jointly learn and predict all predicates and arguments. We extend existing model and introduce unified argument representation with biaffine scorer to the uniform SRL for both span and dependency representation forms. Our model achieves new state-of-the-art results on the CoNLL 2005, 2012 and CoNLL 2008, 2009 benchmarks. Our results show that span and dependency SRL can be effectively handled in a uniform fashion, which for the first time enables us to conveniently explore the useful connection between two types of semantic representation forms.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Björkelund, A.; Bernd, B.; Hafdell, L.; and Nugues, P. 2010. A high-performance syntactic and semantic dependency parser. In *COLING*.

Cai, J.; He, S.; Li, Z.; and Zhao, H. 2018. A full end-to-end semantic role labeler, syntactic-agnostic over syntactic-aware? In *COLING*.

Carreras, X., and Màrquez, L. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *CoNLL*.

Chiu, J. P., and Nichols, E. 2016. Named entity recognition with bidirectional LSTM-CNNs. *TACL*.

Dozat, T., and Manning, C. D. 2017. Deep biaffine attention for neural dependency parsing. In *ICLR*.

FitzGerald, N.; Täckström, O.; Ganchev, K.; and Das, D. 2015. Semantic role labeling with neural network factors. In *EMNLP*.

Gal, Y., and Ghahramani, Z. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *NIPS*.

Gildea, D., and Jurafsky, D. 2002. Automatic labeling of semantic roles. *Computational linguistics*.

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 *CoNLL*.

He, L.; Lee, K.; Lewis, M.; and Zettlemoyer, L. 2017. Deep semantic role labeling: What works and what's next. In *ACL*.

He, L.; Lee, K.; Levy, O.; and Zettlemoyer, L. 2018a. Jointly predicting predicates and arguments in neural semantic role labeling. In *ACL*.

He, S.; Li, Z.; Zhao, H.; Bai, H.; and Liu, G. 2018b. Syntax for semantic role labeling, to be, or not to be. In *ACL*.

Johansson, R., and Nugues, P. 2008a. Dependency-based semantic role labeling of PropBank. In *EMNLP*.

Johansson, R., and Nugues, P. 2008b. Dependency-based syntactic-semantic analysis with PropBank and NomBank. In *CoNLL*.

Kingma, D., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Lee, K.; He, L.; Lewis, M.; and Zettlemoyer, L. 2017. End-to-end neural coreference resolution. In *EMNLP*.

Li, Z.; Cai, J.; He, S.; and Zhao, H. 2018a. Seq2seq dependency parsing. In *COLING*.

Li, Z.; He, S.; Cai, J.; Zhang, Z.; Zhao, H.; Liu, G.; Li, L.; and Si, L. 2018b. A unified syntax-aware framework for semantic role labeling. In *EMNLP*.

Li, Z.; He, S.; Zhang, Z.; and Zhao, H. 2018c. Joint learning of pos and dependencies for multilingual universal dependency parsing. *CoNLL*.

Marcheggiani, D., and Titov, I. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *EMNLP*.

Marcheggiani, D.; Frolov, A.; and Titov, I. 2017. A simple and accurate syntax-agnostic neural model for dependency-based semantic role labeling. In *CoNLL*.

Nair, V., and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*.

Palmer, M.; Gildea, D.; and Kingsbury, P. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*.

Peng, H.; Thomson, S.; Swayamdipta, S.; and Smith, N. A. 2018. Learning joint semantic parsers from disjoint data. In *NAACL*.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *NAACL: HLT*.

Pradhan, S.; Ward, W.; Hacioglu, K.; Martin, J.; and Jurafsky, D. 2005. Semantic role labeling using different syntactic views. In *ACL*.

Pradhan, S.; Moschitti, A.; Xue, N.; Ng, H. T.; Björkelund, A.; Uryupina, O.; Zhang, Y.; and Zhong, Z. 2013. Towards robust linguistic analysis using ontonotes. In *CoNLL*.

Punyakanok, V.; Roth, D.; and Yih, W.-t. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*.

Roth, M., and Lapata, M. 2016. Neural semantic role labeling with dependency path embeddings. In *ACL*.

Strubell, E.; Verga, P.; Andor, D.; Weiss, D.; and Mccallum, A. 2018. Linguistically-informed self-attention for semantic role labeling. *arXiv preprint*.

Surdeanu, M.; Johansson, R.; Meyers, A.; Màrquez, L.; and Nivre, J. 2008. The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL*.

Tan, Z.; Wang, M.; Xie, J.; Chen, Y.; and Shi, X. 2018. Deep semantic role labeling with self-attention. In *AAAI*.

Toutanova, K.; Haghighi, A.; and Manning, C. D. 2008. A global joint model for semantic role labeling. *Computational Linguistics*.

Wang, R.; Zhao, H.; Ploux, S.; Lu, B.-L.; and Utiyama, M. 2016. A bilingual graph-based semantic model for statistical machine translation. In *IJCAI*.

Zhang, Z.; Wu, Y.; Li, Z.; He, S.; Zhao, H.; Zhou, X.; and Zhou, X. 2018. I know what you want: Semantic learning for text comprehension. *arXiv preprint arXiv:1809.02794*.

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 *CoNLL*.

Zhao, H.; Chen, W.; and Kit, C. 2009. Semantic dependency parsing of NomBank and PropBank: An efficient integrated approach via a large-scale feature selection. In *EMNLP*.

Zhao, H.; Zhang, X.; and Kit, C. 2013. Integrative semantic dependency parsing via efficient large-scale feature selection. *Journal of Artificial Intelligence Research*.

Zhou, J., and Xu, W. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *ACL*.