# A Full End-to-End Semantic Role Labeler, Syntax-agnostic Over Syntax-aware?

**Jiaxun Cai**[1,2], **Shexia He**[1,2], **Zuchao Li**[1,2], **Hai Zhao**[1,2]*
[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
{caijiaxun, heshexia, charlee}@sjtu.edu.cn,
zhaohai@cs.sjtu.edu.cn

## Abstract

Semantic role labeling (SRL) is to recognize the predicate-argument structure of a sentence, including subtasks of predicate disambiguation and argument labeling. Previous studies usually formulate the entire SRL problem into two or more subtasks. For the first time, this paper introduces an end-to-end neural model which unifiedly tackles the predicate disambiguation and the argument labeling in one shot. Using a biaffine scorer, our model directly predicts all semantic role labels for all given word pairs in the sentence without relying on any syntactic parse information. Specifically, we augment the BiLSTM encoder with a non-linear transformation to further distinguish the predicate and the argument in a given sentence, and model the semantic role labeling process as a word pair classification task by employing the biaffine attentional mechanism. Though the proposed model is syntax-agnostic with local decoder, it outperforms the state-of-the-art syntax-aware SRL systems on the CoNLL-2008, 2009 benchmarks for both English and Chinese. To our best knowledge, we report the first syntax-agnostic SRL model that surpasses all known syntax-aware models.

## 1 Introduction

Semantic role labeling (SRL) is a shallow semantic parsing, which is dedicated to identifying the semantic arguments of a predicate and labeling them with their semantic roles. SRL is considered as one of the core tasks in the natural language processing (NLP), which has been successfully applied to various downstream tasks, such as information extraction (Bastianelli et al., 2013), question answering (Shen and Lapata, 2007; Berant et al., 2013), machine translation (Xiong et al., 2012; Shi et al., 2016).

Typically, SRL task can be put into two categories: constituent-based (i.e., phrase or span) SRL and dependency-based SRL. This paper will focus on the latter one popularized by CoNLL-2008 and 2009 shared tasks (Surdeanu et al., 2008; Hajič et al., 2009). Most conventional SRL systems relied on sophisticated handcraft features or some declarative constraints (Pradhan et al., 2005; Zhao et al., 2009a), which suffers from poor efficiency and generalization ability. A recently tendency for SRL is adopting neural networks methods attributed to their significant success in a wide range of applications (Bai and Zhao, 2018; Zhang and Zhao, 2018). However, most of those works still heavily resort to syntactic features. Since the syntactic parsing task is equally hard as SRL and comes with its own errors, it is better to get rid of such prerequisite as in other NLP tasks. Accordingly, Marcheggiani et al. (2017) presented a neural model putting syntax aside for dependency-based SRL and obtain favorable results, which overturns the inherent belief that syntax is indispensable in SRL task (Punyakanok et al., 2008).

Besides, SRL task is generally formulated as multi-step classification subtasks in pipeline systems, consisting of predicate identification, predicate disambiguation, argument identification and argument classification. Most previous SRL approaches adopt a pipeline framework to handle these subtasks one

*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).

*Proceedings of the 27th International Conference on Computational Linguistics*, pages 2753–2765
Santa Fe, New Mexico, USA, August 20-26, 2018.

after another. Until recently, some works (Zhou and Xu, 2015; He et al., 2017) introduce end-to-end models for span-based SRL, which motivates us to explore integrative model for dependency SRL.

In this work, we propose a syntactic-agnostic end-to-end system, dealing with predicate disambiguation and argument labeling in one model, unlike previous systems that treat the predicate disambiguation as a subtask and handle it separately. In detail, our model contains (1) a deep BiLSTM encoder, which is able to distinguish the predicates and arguments by mapping them into two different vector spaces, and (2) a biaffine attentional (Dozat and Manning, 2017) scorer, which unifiedly predicts the semantic role for argument and the sense for predicate.

We experimentally show that though our biaffine attentional model remains simple and does not rely on any syntactic feature, it achieves the best result on the benchmark for both Chinese and English even compared to syntax-aware systems. In summary, our major contributions are shown as follows:

- We propose an accurate syntax-agnostic model for neural SRL, which outperforms the best reported syntax-aware model, breaking the long-held belief that syntax is a prerequisite for SRL.
- Our model gives state-of-the-art results on the CoNLL-2008, CoNLL-2009 English and Chinese benchmarks, scoring 85.0% $F_1$, 89.6% $F_1$ and 84.4% $F_1$, respectively.
- Our work is the first attempt to apply end-to-end model for dependency-based SRL, which tackles the predicate disambiguation and the argument labeling subtasks in one shot.

## 2 Semantic Structure Decomposition

SRL includes two subtasks: predicate identification/disambiguation and argument identification/labeling. Since the CoNLL-2009 dataset provides the gold predicates, most previous neural SRL systems use a default model to perform predicate disambiguation and focus on argument identification/labeling. Despite nearly all SRL work adopted the pipeline model with two or more components, Zhao and Kit (2008) and Zhao et al. (2013) presented an end-to-end solution for the entire SRL task with a word pair classifier. Following the same formulization, we propose the first neural SRL system that uniformly handles the tasks of predicate disambiguation and argument identification/labeling.

In semantic dependency parsing, we can always identify two types of words, semantic head (predicate) and semantic dependent (argument). To build the needed predicate-argument structure, the model only needs to predict the role of any word pair from the given sentence. For the purpose, an additional role label *None* and virtual root node <*VR*> are introduced. The *None* label indicates that there is no semantic role relationship inside the word pair. We insert a virtual root <*VR*> in the head of the sentence, and set it as the semantic head of all the predicates. By introducing the *None* label and the <*VR*> node, we construct a semantic tree rooted at the <*VR*> node with several virtual arcs labeled with *None*. Thus, the predicate disambiguation and argument identification/labeling tasks can be naturally regarded as the labeling process over all the word pairs. Figure 1 shows an example of the semantic graph augmented with a virtual root and virtual arc, and Table 1 lists all the corresponding word pair examples, in which two types of word pairs are included, <*VR*> followed by predicate candidates[1] and a known predicate collocated with every words in the sentence as argument candidates. Note that since the nominal predicate sometimes takes itself as its argument, the predicate itself is also included in the argument candidate list.

## 3 Model

Our model contains two main components: (1) a deep BiLSTM encoder that takes each word embedding $e$ of the given sentence as input and generates dense vectors for both words in the to-be-classified word pair respectively, (2) a biaffine attentional scorer which takes the hidden vectors for the given word pair as input and predict a label score vector. Figure 2 provides an overview of our model.

[1] Note that there is a key difference between CoNLL 2008 and 2009 shared task for English, the latter has specified the predicate in the data so that here we have only one sample for <*VR*>-predicate pair to make predicate disambiguation. For the former, predicate candidates should be every words in the given sentence. More details are seen in Section 4.4.

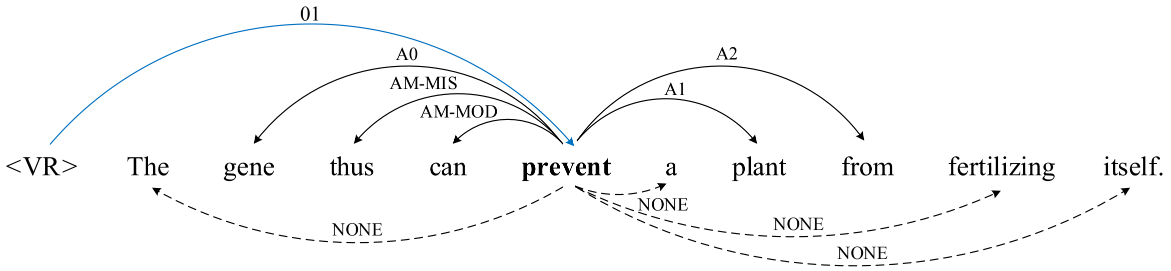

Figure 1: The dependency graph with a virtual node <VR> and virtual arcs with *None* label.

| Head | dependent (word pair) | Label | Head | dependent (word pair) | Label |
|---|---|---|---|---|---|
| <VR> | prevent | 01 | <VR> | fertilizing | 01 |
| prevent | The | None | fertilizing | The | None |
| prevent | gene | A0 | fertilizing | gene | None |
| prevent | thus | AM-MIS | fertilizing | thus | None |
| prevent | can | AM-MOD | fertilizing | can | None |
| prevent | prevent | None | fertilizing | prevent | None |
| … | … | … | … | … | … |
| prevent | itself | None | fertilizing | itself | A1 |

Table 1: Word pairs for semantic role label classification.

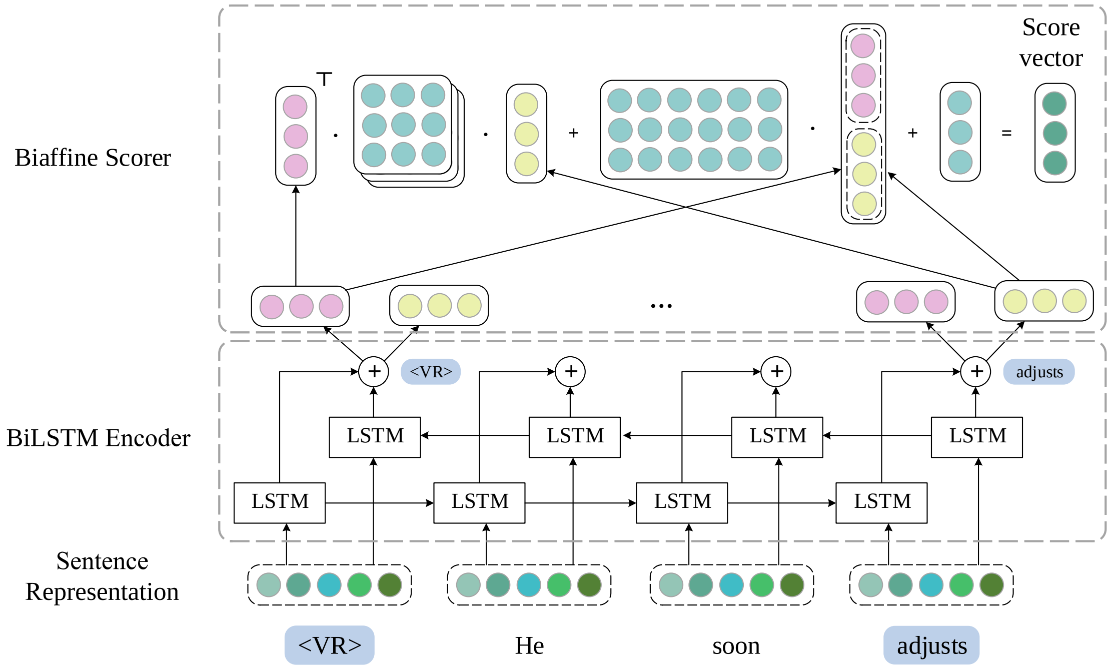

Figure 2: An overview of our model.

### 3.1 Bidirectional LSTM Encoder

**Word Representation** The word representation of our model is the concatenation of several vectors: a randomly initialized word embedding $\boldsymbol{e}^{(r)}$, a pre-trained word embedding $\boldsymbol{e}^{(p)}$, a randomly initialized part-of-speech (POS) tag embedding $\boldsymbol{e}^{(pos)}$, a randomly initialized lemma embedding $\boldsymbol{e}^{(l)}$. Besides, since previous work (He et al., 2018) demonstrated that the predicate-specific feature is helpful in promoting the role labeling process, we employ an indicator embedding $\boldsymbol{e}^{(i)}$ to indicate whether a word is a predicate when predicting and labeling the arguments for each given predicate. The final word representation is given by $\boldsymbol{e} = \boldsymbol{e}^{(r)} \oplus \boldsymbol{e}^{(p)} \oplus \boldsymbol{e}^{(l)} \oplus \boldsymbol{e}^{(pos)} \oplus \boldsymbol{e}^{(i)}$, where $\oplus$ is the concatenation operator.

**Encoder** As commonly used to model the sequential input in most NLP tasks (Wang et al., 2016; He et al., 2018), BiLSTM is adopted for our sentence encoder. By incorporating a stack of two distinct LSTMs, BiLSTM processes an input sequence in both forward and backward directions. In this way, the BiLSTM encoder provides the ability to incorporate the contextual information for each word.

Given a sequence of word representation $S = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_N\}$ as input, the $i$-th hidden state $\boldsymbol{g}_i$ is encoded as follows:

$$\boldsymbol{g}_i^f = LSTM^{\mathcal{F}}\left(\boldsymbol{e}_i, \boldsymbol{g}_{i-1}^f\right), \quad \boldsymbol{g}_i^b = LSTM^{\mathcal{B}}\left(\boldsymbol{e}_i, \boldsymbol{g}_{i+1}^b\right), \quad \boldsymbol{g}_i = \boldsymbol{g}_i^f \oplus \boldsymbol{g}_i^b,$$

where $LSTM^{\mathcal{F}}$ denotes the forward LSTM transformation and $LSTM^{\mathcal{B}}$ denotes the backward LSTM transformation. $\boldsymbol{g}_i^f$ and $\boldsymbol{g}_i^b$ are the hidden state vectors of the forward LSTM and backward LSTM respectively.

### 3.2 Biaffine Attentional Role Scorer

Typically, to predict and label arguments for a given predicate, a role classifier is employed on top of the BiLSTM encoder. Some work like (Marcheggiani et al., 2017) shows that incorporating the predicate's hidden state in their role classifier enhances the model performance, while we argue that a more natural way to incorporate the syntactic information carried by the predicate is to employ the attentional mechanism. Our model adopts the recently introduced biaffine attention (Dozat and Manning, 2017) to enhance our role scorer. Biaffine attention is a natural extension of bilinear attention (Luong et al., 2015) which is widely used in neural machine translation (NMT).

**Nonlinear Affine Transformation** Usually, a BiLSTM decoder takes the concatenation $\boldsymbol{g}_i$ of the hidden state vectors as output for each hidden state. However, in the SRL context, the encoder is supposed to distinguish the currently considered predicate from its candidate arguments. To this end, we perform two distinct affine transformations with a nonlinear activation on the hidden state $\boldsymbol{g}_i$, mapping it to vectors with smaller dimensionality:

$$\boldsymbol{h}_i^{(pred)} = ReLU\left(\boldsymbol{W}^{(pred)}\boldsymbol{g}_i + \boldsymbol{b}^{(pred)}\right), \;\; \boldsymbol{h}_i^{(arg)} = ReLU\left(\boldsymbol{W}^{(arg)}\boldsymbol{g}_i + \boldsymbol{b}^{(arg)}\right),$$

where $ReLU$ is the rectilinear activation function, $\boldsymbol{h}_i^{(pred)}$ is the hidden representation for the predicate and $\boldsymbol{h}_i^{(arg)}$ is the hidden representation for the candidate arguments.

By performing such transformations over the encoder output to feed the scorer, the latter may benefit from deeper feature extraction. First, ideally, instead of keeping both features learned by the two distinct LSTMs, the scorer is now enabled to learn features composed from both recurrent states together with reduced dimensionality. Second, it provides the ability to map the predicates and the arguments into two distinct vector spaces, which is essential for our tasks since some words can be labeled as predicate and argument simultaneously. Mapping a word into two different vectors can help the model disambiguate the role that it plays in different context.

**Biaffine Scoring** In the standard NMT context, given a target recurrent output vector $h_i^{(t)}$ and a source recurrent output vector $h_j^{(s)}$, a bilinear transformation calculates a score $s_{ij}$ for the alignment:

$$s_{ij} = \boldsymbol{h}_i^{\top(t)}\boldsymbol{W}\boldsymbol{h}_j^{(s)},$$

However, considering that in a traditional classification task, the distribution of classes is often uneven, and that the output layer of the model normally includes a bias term designed to capture the prior probability $P(y_i = c)$ of each class, with the rest of the model focusing on learning the likelihood of each class given the data $P(y_i = c|x_i)$, (Dozat and Manning, 2017) introduced the bias terms into the bilinear attention to address such uneven problem, resulting in a biaffine transformation. The biaffine transformation is a natural extension of the bilinear transformation and the affine transformation. In SRL task, the distribution of the role labels is similarly uneven and the problem comes worse after we introduce the additional <*VR*> node and *None* label, directly applying the primitive form of bilinear attention would fail to capture the prior probability $P(y_i = c_k)$ for each class. Thus, the biaffine attention introduced in our model would be extremely helpful for semantic role prediction.

It is worth noting that in our model, the scorer aims to assign a score for each specific semantic role. Besides learning the prior distribution for each label, we wish to further capture the preferences for the label that a specific predicate-argument pair can take. Thus, our biaffine attention contains two distinguish bias terms:

$$\boldsymbol{s}_{ij} = \boldsymbol{h}_i^{\top(arg)} \boldsymbol{W}^{(role)} \boldsymbol{h}_j^{(pred)} \tag{1}$$

$$+ \boldsymbol{U}^{(role)} \left( \boldsymbol{h}_i^{(arg)} \oplus \boldsymbol{h}_j^{(pred)} \right) \tag{2}$$

$$+ \boldsymbol{b}^{(role)}, \tag{3}$$

where $\boldsymbol{W}^{(role)}$, $\boldsymbol{U}^{(role)}$ and $\boldsymbol{b}^{(role)}$ are parameters that will be updated by some gradient descent methods in the learning process. There are several points that should be paid attention to in the above biaffine transformation. First, since our goal is to predict the label for each pair of $\boldsymbol{h}_i^{(arg)}$, $\boldsymbol{h}_j^{(pred)}$, the output of our biaffine transformation should be a vector of dimensionality $N_r$ instead of a real value, where $N_r$ is the number of all the candidate semantic labels. Thus, the bilinear transformation in Eq. (1) maps two input vectors into another vector. This can be accomplished by setting $\boldsymbol{W}^{(role)}$ as a $(d_h \times N_r \times d_h)$ matrix, where $d_h$ is the dimensionality of the hidden state vector. Similarly, the output of the linear transformation in Eq. (2) is also a vector by setting $\boldsymbol{U}^{(role)}$ as a $(N_r \times 2d_h)$ matrix. Second, Eq. (2) captures the preference of each role (or sense) label condition on taking the $j$-th word as predicate and the $i$-th word as argument. Third, the last term $\boldsymbol{b}^{(role)}$ captures the prior probability of each class $P(y_i = c_k)$. Notice that Eq. (2) and (3) capture different kinds of bias for the latent distribution of the label set.

Given a sentence of length $L$ (including the <*VR*> node), for one of its predicates $w_j$, the scorer outputs a score vector $\{\boldsymbol{s}_{1j}, \boldsymbol{s}_{2j}, \cdots, \boldsymbol{s}_{Lj}\}$. Then our model picks as its output the label with the highest score from each score vector: $y_{ij} = \arg\max_{1 \le k \le N_r}(\boldsymbol{s}_{ij}[k])$, where $\boldsymbol{s}_{ij}[k]$ denotes the score of the $k$-th candidate semantic label.

## 4 Experiments

### 4.1 Dataset and Training Detail

We evaluate our model[2] on English and Chinese CoNLL-2009 datasets with the standard split into training, test and development sets. The pre-trained embedding for English is trained on Wikipedia and Gigaword using the GloVe (Pennington et al., 2014), while those for Chinese is trained on Wikipedia. Our implementation uses the DyNet[3] library for building the dynamic computation graph of the network.

When not otherwise specified, our model uses: 100-dimensional word, lemma, pre-trained and POS tag embeddings and 16-dimensional predicate-specific indicator embedding; and a 20% chance of dropping on the whole word representation; 3-layer BiLSTMs with 400-dimensional forward and backward LSTMs, using the form of recurrent dropout suggested by (Gal and Ghahramani, 2016) with an 80% keep probability between time-steps and layers; two 300-dimensional affine transformation with the ReLU non-linear activation on the output of BiLSTM, also with an 80% keep probability.

[2] The code is available at `https://github.com/JiaxunCai/Dynet-Biaffine-SRL`

[3] https://github.com/clab/dynet

| *Syntax-aware system (single)* | P | R | F$_1$ |
|---|---|---|---|
| Zhao et al. (2009a) | – | – | 86.2 |
| Zhao et al. (2009c) | – | – | 85.4 |
| Björkelund et al. (2010) | 87.1 | 84.5 | 85.8 |
| Lei et al. (2015) | – | – | 86.6 |
| 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 |
| **He et al. (2018)** | **89.7** | **89.3** | **89.5** |
| *Syntax-aware system (ensemble)* | P | R | F$_1$ |
| Roth and Lapata (2016) | 90.3 | 85.7 | 87.9 |
| Marcheggiani and Titov (2017) | 90.5 | 87.7 | 89.1 |
| *Syntax-agnostic system* | P | R | F$_1$ |
| Marcheggiani et al. (2017) | 88.7 | 86.8 | 87.7 |
| He et al. (2018) | 89.5 | 87.9 | 88.7 |
| **This work** | **89.9** | **89.2** | **89.6** |

Table 2: Results on English in-domain (WSJ) test set.

The parameters in our model are optimized with Adam (Kingma and Ba, 2015), which keeps a moving average of the L2 norm of the gradient for each parameter throughout training and divides the gradient for each parameter by this moving average, ensuring that the magnitude of the gradients will on average be close to one. For the parameters of optimizer, we follow the settings in (Dozat and Manning, 2017), with $\beta_1 = \beta_2 = 0.9$ and learning rate $0.002$, annealed continuously at a rate of $0.75$ every $5,000$ iterations, with batches of approximately $5,000$ tokens. The maximum number of epochs of training is set to 50.

### 4.2 Results

Tables 2 and 3 report the comparison of performance between our model and previous dependency-based SRL model on both English and Chinese. Note that the predicate disambiguation subtask is unifiedly tackled with arguments labeling in our model with precisions of 95.0% and 95.6% respectively on English and Chinese test sets in our experiments[4]. The proposed model accordingly outperforms all the SRL systems so far on both languages, even including those syntax-aware and ensemble ones. The improvement grows even larger when comparing only with the single syntax-agnostic models.

For English, our syntax-agnostic model even slightly outperforms the best reported syntax-aware model (He et al., 2018) with a margin of 0.1% F$_1$. Compared to syntax-agnostic models, our model overwhelmingly outperforms (with an improvement of 0.9% F$_1$) the previous work (He et al., 2018).

Although we used the same parameters as for English, our model substantially outperforms the state-of-art models on Chinese, demonstrating that our model is robust and less sensitive to the parameter selection. For Chinese, the proposed model outperforms the best previous model (He et al., 2018) with a considerable improvement of 1.5% F$_1$, and surpasses the best single syntax-agnostic model (He et al., 2018) with a margin of 2.5% F$_1$.

Table 4 compares the results on English out-of-the-domain (Brown) test set, from which our model still remains strong. The proposed model gives a comparable result with the highest score from syntax-aware model of (He et al., 2018), which affirms that our model does well learn and generalize the latent semantic preference of the data.

Results on both in-domain and out-of-the-domain test sets demonstrate the effectiveness and the robustness of the proposed model structure—the non-linear transformation after the BiLSTM serves to distinguish the predicate from argument while the biaffine attention tells what to attend for each candidate argument. In Section 4.3.2, we will get an insight into our model and explore how each individual

[4] Note that we give comparable predicate disambiguation results with He et al. (2018), with 95.01% and 95.58% F$_1$ on development and test sets, respectively.

| *Syntax-aware system* | P | R | $F_1$ |
|---|---|---|---|
| Zhao et al. (2009a) | 80.4 | 75.2 | 77.7 |
| Björkelund et al. (2009) | 82.4 | 75.1 | 78.6 |
| Roth and Lapata (2016) | 83.2 | 75.9 | 79.4 |
| Marcheggiani and Titov (2017) | 84.6 | 80.4 | 82.5 |
| **He et al. (2018)** | **84.2** | **81.5** | **82.8** |
| *Syntax-agnostic system* | P | R | $F_1$ |
| Marcheggiani et al. (2017) | 83.4 | 79.1 | 81.2 |
| He et al. (2018) | 84.5 | 79.3 | 81.8 |
| **This work** | **84.7** | **84.0** | **84.3** |

Table 3: Results on Chinese in-domain test set.

| *Syntax-aware system* | P | R | $F_1$ |
|---|---|---|---|
| Björkelund et al. (2010) | 75.7 | 72.2 | 73.9 |
| Lei et al. (2015) | – | – | 75.6 |
| FitzGerald et al. (2015) | – | – | 75.2 |
| Roth and Lapata (2016) | 76.9 | 73.8 | 75.3 |
| Marcheggiani and Titov (2017) | 78.5 | 75.9 | 77.2 |
| **He et al. (2018)** | **81.9** | **76.9** | **79.3** |
| *Syntax-agnostic system* | P | R | $F_1$ |
| Marcheggiani et al. (2017) | 79.4 | 76.2 | 77.7 |
| He et al. (2018) | 81.7 | 76.1 | 78.8 |
| **This work** | **79.8** | **78.3** | **79.0** |

Table 4: Results on English out-of-the-domain (Brown) test set.

|  | **AL** |  |  | **PD** |
|---|---|---|---|---|
| System | P | R | $F_1$ | P |
| without POS | 89.5 | 89.1 | 89.3 | 94.9 |
| without lemma | 89.5 | 89.3 | 89.4 | 94.9 |
| without indicator | 89.1 | 88.5 | 88.8 | 95.0 |
| **This work** | **89.9** | **89.2** | **89.6** | **95.0** |

Table 5: Contribution of the input representation. Acronyms used: **AL**-argument labeling, **PD**-predicate disambiguation.

component impacts the model performance.

## 4.3 Ablation Analysis

### 4.3.1 Word Representation

To learn how the input word representation choice impacts our model performance, we conduct an ablation study on the English test set whose results are shown in Table 5. Since we deal with the two subtasks in a single model, the choice of word representation will simultaneously influence the results of both of them. Besides the results of argument labeling, we also report the precision of predicate disambiguation.

The results demonstrate that the multiple dimensional indicator embedding proposed by (He et al., 2018) contributes the most to the final performance of our model. It is consistent with the conclusion in (Marcheggiani et al., 2017) which argue that encoding predicate information promotes the SRL model. It is interesting that the impact of POS tag embedding (about 0.3% $F_1$) is less compared to the previous works, which possibly allows us to build an accuracy model even when the POS tag label is unavailable.

### 4.3.2 Into the Model

In this section, we get insight into the proposed model, exploring how the deep BiLSTM encoder and the biaffine attention affect the labeling results respectively. Specifically, we present two groups of results on the CoNLL-2009 English test set. 1) Shallow biaffine attentive (SBA) labeler. Instead of mapping the output of the BiLSTM into two distinct vector spaces, we apply a single non-linear affine transformation on the output. The single transformation just serves to reduce the dimensionality and does not differ the predicates from the arguments. 2) Deep bilinear attentive (DBA) labeler. We apply the primitive form of bilinear attention in the scorer by removing the two bias terms of the biaffine transformation. By this means, we learn to what extent can the bias terms fit the prior distribution of the data. Results of the two experiments are shown in Table 6.

The results show that the bias terms in biaffine attention play an important role in promoting the model performance. Removal of the bias terms dramatically declines the performance by 1.7% $F_1$. Thus we can draw a conclusion that the bias term does well in fitting the prior distribution and global preference

| | **AL** | | | **PD** |
|---|---|---|---|---|
| System | P | R | $F_1$ | P |
| SBA-labeler | 89.5 | 88.8 | 89.1 | 94.7 |
| DBA-labeler | 88.1 | 87.7 | 87.9 | 94.3 |
| **This work** | **89.9** | **89.2** | **89.6** | **95.0** |

Table 6: Contribution of the model components.

| | **AL** | | | **PD** |
|---|---|---|---|---|
| System | P | R | $F_1$ | P |
| *with* pruning | 88.2 | 85.6 | 86.9 | 95.0 |
| *without* pruning | **89.9** | **89.2** | **89.6** | **95.0** |

Table 7: Comparison of results with and without argument candidate pruning.

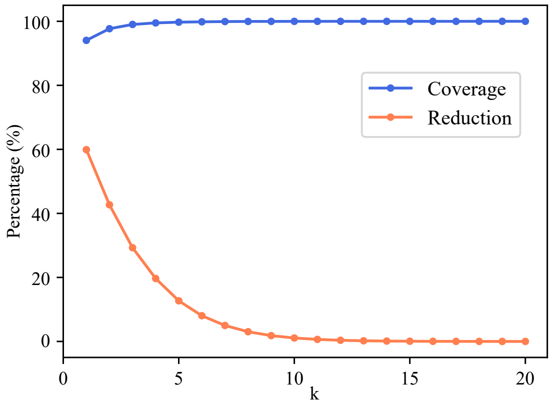

Figure 3: Coverage rate of true arguments and reduction rate of argument candidates against the pruning order $k$ on the English training set.

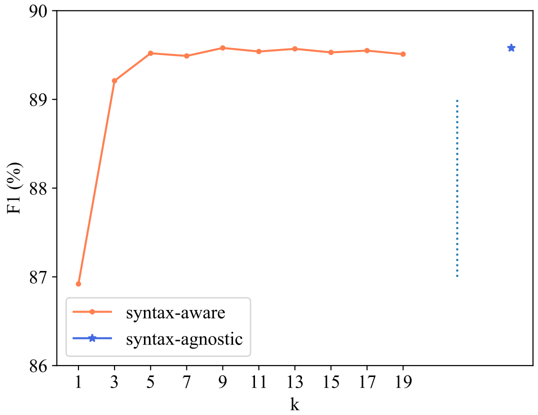

Figure 4: F1 scores against the pruning order $k$ on English test set.

of the data. The bilinear attentional model behaves more poorly since it struggles to learn the likelihood of each class on an uneven data set without knowledge about the prior distribution. Though the deep encoder contributes less to the performance, it also brings an improvement of 0.5% $F_1$. Note that the only difference of SBA-labeler of our standard model is whether the hidden representations of the arguments and the predicates lay in different vector spaces. Such a result confirms that distinguishing the predicates from the arguments in encoding process indeed enhances the model to some extent.

#### 4.3.3 Syntax-aware and Syntax-agnostic

Noting that the work (Zhao and Kit, 2008) and (Zhao et al., 2013) are similar to ours in modeling the dependency-based SRL tasks as word pair classification, and that they successfully incorporate the syntactic information by applying argument candidate pruning, we further perform empirical study to explore whether employing such pruning method enhance or hinder our model. Specifically, we use the automatically predicted parse with moderate performance provided by CoNLL-2009 shared task, with the LAS score about 86%.

The pruning method is supposed to work since it can alleviate the imbalanced label distribution caused by introducing the *None* label. However, as shown in Table 7, the result is far from satisfying. The main reason might be the pruning algorithm is so strict that too many true arguments are falsely pruned. To address this problem, He et al. (2018) introduced an extended $k$-order argument pruning algorithm. Figure 3 shows the curves of coverage and reduction rate against the pruning order $k$ on the English training set following (He et al., 2018). Following this work, we further perform different orders of pruning and obtain the $F_1$ scores curve shown in Figure 4. However, the $k$-order pruning does not boost the performance of our model. Table 8 presents the performance gap between syntax-agnostic and syntax-aware settings of the same models. Unlike the other two works, the introduction of syntax information fails to bring about bonus for our model. Nevertheless, it is worth noting that even when running without the syntax information, our mode still show a promising result compared to the other syntax-aware models.

| System | syntax-agnostic | syntax-aware | Δ $F_1$ |
|---|---|---|---|
| Marcheggiani and Titov (2017) | 87.7 | 88.0 | 0.3 |
| He et al. (2018) (CoNLL-2009 predicted) | 88.7 | 89.5 | 0.8 |
| He et al. (2018) (gold syntax) | 88.7 | 90.3 | 1.6 |
| This work (CoNLL-2009 predicted) | 89.6 | 89.6 | ≈0 |

Table 8: The absolute performance gaps between syntax-agnostic and syntax-aware settings. Both (He et al., 2018) and our models use 10-order pruning according to syntactic parse tree.

### 4.4 CoNLL 2008: Augment the Model with Predicate Identification

Though CoNLL-2009 provided the gold predicate beforehand, the predicate identification subtask is still indispensable for a real world SRL task. Thus, we further augment our model with the predicate identification ability.

Specifically, we first attach all the words in the sentence to the virtual root <*VR*> and label the word which is not a predicate with the *None* role label. It should be noting that, in CoNLL-2009 settings, we just attach the predicates to the virtual root, since we do not need to distinguish the predicate from other word. The training scheme still keeps the same as that in CoNLL-2009 settings, while in testing phase, an additional procedure is performed to find out all the predicates of a given sentence.

First, our model is fed the representations of the virtual root and each word of the input sentence, identifying and disambiguating all the predicates of the sentence. Second, it picks each predicate predicted by the model with each word of the sentence to identify and label the semantic role in between, which remains the same as the model does on CoNLL-2009. The second phase is repeated until all the predicates have got its arguments being identified and labeled. We evaluate our model on CoNLL-2008 benchmark using the same hyperparameters settings mentioned in Section 4.1 except that we remove the predicate-specific indicator feature.

The $F_1$ scores on predicates identification and labeling of our model is 89.43%, which remain comparable with the most recent work (He et al., 2018) (90.53% $F_1$). As shown in Table 9, though tackling all the subtasks of CoNLL-2008 SRL unifiedly in a full end-to-end manner, our model outperforms the best reported results with a large margin of about 1.7% semantic $F_1$.

| | | Predicate Labeling | | | Semantic Labeling | | |
|---|---|---|---|---|---|---|---|
| System | LAS | P | R | $F_1$ | P | R | $F_1$ |
| Johansson and Nugues (2008) | 90.13 | – | – | – | – | – | 81.75 (80.37) |
| Zhao and Kit (2008) | 87.52 | – | – | – | 80.57 | 74.97 | 77.67 |
| Zhao et al. (2009b) | 88.39 | – | – | – | – | – | 82.1 (80.53) |
| | 89.28 | – | – | – | – | – | 82.5 (80.94) |
| Zhao et al. (2013) | 88.39 | – | – | (87.15) | – | – | 82.5 (80.91) |
| | 89.28 | – | – | (86.47) | – | – | 82.4 (80.88) |
| He et al. (2018) (syntax-aware) | 86.0 | 89.73 | 91.35 | 90.53 | 83.9 | 82.7 | 83.3 |
| He et al. (2018) (syntax-agnostic) | – | 89.73 | 91.35 | 90.53 | 83.5 | 82.4 | 82.9 |
| **Ours (syntax-agnostic)** | – | **88.9** | **90.0** | **89.4 (87.9)** | **84.7** | **85.2** | **85.0 (83.6)** |

Table 9: Results on the CoNLL-2008 test set (WSJ). The results enclosed with parenthesis are evaluated on WSJ + Brown test set, following the official evaluation setting of CoNLL-2008 shared task.

## 5 Related Work

Semantic role labeling was pioneered by Gildea and Jurafsky (2002). Most traditional SRL models heavily rely on complex feature engineering (Pradhan et al., 2005; Zhao et al., 2009a; Björkelund et al., 2009). Among those early works, Pradhan et al. (2005) combined features derived from different syntactic parses based on SVM classifier, while Zhao et al. (2009a) exploited the abundant set of language-specific features that were carefully designed for SRL task.

In recent years, applying neural networks in SRL task has gained a lot of attention due to the impressive success of deep neural networks in various NLP tasks (Zhang et al., 2016; Cai et al., 2017; Qin et al., 2017; Cai and Zhao, 2017). Collobert et al. (2011) initially introduced neural networks into the SRL task. They developed a feed-forward network that employed a convolutional network as sentence encoder and a conditional random field as a role classifier. Foland and Martin (2015) extended their model to further use syntactic information by including binary indicator features. FitzGerald et al. (2015) exploited a neural network to unifiedly embed arguments and semantic roles, similar to the work (Lei et al., 2015) which induced a compact feature representation applying tensor-based approach. Roth and Lapata (2016) introduced the dependency path embedding to incorporate syntax and exhibited a notable success, while Marcheggiani and Titov (2017) employed the graph convolutional network to integrate syntactic information into their neural model.

Besides the above-mentioned works who relied on syntactic information, several works attempted to build SRL systems without or with little syntactic information. Zhou and Xu (2015) came up with an end-to-end model for span-based SRL and obtained surprising performance putting syntax aside. He et al. (2017) further extended their work with the highway network. Simultaneously, Marcheggiani et al. (2017) proposed a syntax-agnostic model with effective word representation for dependency-based SRL.

However, almost all of previous works treated the predicate disambiguation as individual subtasks, apart from (Zhao and Kit, 2008; Zhao et al., 2009a; Zhao et al., 2009c; Zhao et al., 2013), who presented the first end-to-end system for dependency SRL. For the neural models of dependency SRL, we have presented the first end-to-end solution that handles both semantic labeling subtasks in one single model. At the same time, our model enjoys the advantage that does not rely any syntactic information.

This work is also closely related to the attentional mechanism. The traditional attention mechanism was proposed by Bahdanau et al. (2015) in the NMT literature. Following the work (Luong et al., 2015) that encouraged substituting the MLP in the attentional mechanism with a single bilinear transformation, Dozat and Manning (2017) introduced the bias terms into the primitive form of bilinear attention and applied it for dependency parsing. They demonstrate that the bias terms help their model to capture the uneven prior distribution of the data, which is again verified by our practice on SRL in this paper.

Different from the latest strong syntax-agnostic models in (Marcheggiani and Titov, 2017) and (He et al., 2018) which both adopted sequence labeling formulization for the SRL task, this work adopts word pair classification scheme implemented by LSTM encoder and biaffine scorer. Compared to the previous state-of-the-art syntax-agnostic model in (He et al., 2018) whose performance boosting (more than 1% absolute gain) is mostly due to introducing the enhanced representation, namely, the CNN-BiLSTM character embedding from (Peters et al., 2018), our performance promotion mainly roots from model architecture improvement, which results in quite different syntax-aware enhanced impacts. Using the same latest syntax-aware $k$-order pruning, the syntax-agnostic backbone in (He et al., 2018) may receive about 1% performance gain, while our model is furthermore enhanced little. This comparison also suggests the possibility that maybe our model can be further improved by incorporating with the same character embedding as (He et al., 2018) does[5].

## 6 Conclusion and Future Work

This paper presents a full end-to-end neural model for dependency-based SRL. It is the first time that a SRL model shows its ability to unifiedly handle the predicate disambiguation and the argument labeling subtasks. Our model is effective while remains simple. Experiments show that it achieves the best scores on CoNLL benchmark both for English and Chinese, outperforming the previous state-of-the-art models even with syntax-aware features. Our further investigation by incorporating the latest syntax-aware pruning algorithm shows that the proposed model is insensitive to the input syntactic information, demonstrating an interesting performance style for the SRL task. Of course, we cannot exclude the possibility that the proposed model can be furthermore improved by other syntactic information integration ways, which is left for the future work.

[5] Such an attempt may be hindered by too luxurious computational resource requirement, as there comes extremely high graphic memory prerequisite when integrating both biaffine scorer and the ELMo character embedding.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of 3rd International Conference on Learning Representations (ICLR)*.

Hongxiao Bai and Hai Zhao. 2018. Deep enhanced representation for implicit discourse relation recognition. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*.

Emanuele Bastianelli, Giuseppe Castellucci, Danilo Croce, and Roberto Basili. 2013. Textual inference and meaning representation in human robot interaction. In *Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora*, pages 65–69.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1533–1544, Seattle, Washington, USA, October.

Anders Björkelund, Love Hafdell, and Pierre Nugues. 2009. Multilingual semantic role labeling. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 43–48, Boulder, Colorado, June.

Anders Björkelund, Bohnet Bernd, Love Hafdell, and Pierre Nugues. 2010. A high-performance syntactic and semantic dependency parser. In *Proceedings of the 23rd International Conference on Computational Linguistics (CoLING 2010)*, pages 33–36, Beijing, China, August.

Deng Cai and Hai Zhao. 2017. *Pair-Aware Neural Sentence Modeling for Implicit Discourse Relation Classification*. IEA/AIE 2017, Part II, LNAI 10351.

Deng Cai, Hai Zhao, Zhisong Zhang, Yuan Xin, Yongjian Wu, and Feiyue Huang. 2017. Fast and accurate neural word segmentation for Chinese. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 608–615.

Ronan Collobert, Jason Weston, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(1):2493–2537.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of 5th International Conference on Learning Representations (ICLR)*.

Nicholas FitzGerald, Oscar Tckstrm, Kuzman Ganchev, and Dipanjan Das. 2015. Semantic role labeling with neural network factors. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 960–970.

William Foland and James Martin. 2015. Dependency-based semantic role labeling using convolutional neural networks. In *Joint Conference on Lexical and Computational Semantics*, pages 279–288.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: representing model uncertainty in deep learning. In *International Conference on International Conference on Machine Learning (ICML)*, pages 1050–1059.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado, June.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 473–483, Vancouver, Canada, July.

Shexia He, Zuchao Li, Hai Zhao, Hongxiao Bai, and Gongshen Liu. 2018. Syntax for semantic role labeling, to be, or not to be. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*.

Richard Johansson and Pierre Nugues. 2008. Dependency-based syntactic-semantic analysis with propbank and nombank. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning (CoNLL)*, pages 183–187.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization.

Tao Lei, Yuan Zhang, Llus Mrquez, Alessandro Moschitti, and Regina Barzilay. 2015. High-order low-rank tensors for semantic role labeling. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL: HLT)*, pages 1150–1160.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421, Lisbon, Portugal, September.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1506–1515, Copenhagen, Denmark, September.

Diego Marcheggiani, Anton Frolov, and Ivan Titov. 2017. A simple and accurate syntax-agnostic neural model for dependency-based semantic role labeling. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 411–420, Vancouver, Canada, August.

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

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL: HLT)*, New Orleans, Louisiana.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James Martin, and Daniel Jurafsky. 2005. Semantic role labeling using different syntactic views. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 581–588, Ann Arbor, Michigan, June.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

Lianhui Qin, Zhisong Zhang, Hai Zhao, Zhiting Hu, and Eric P. Xing. 2017. Adversarial connective-exploiting networks for implicit discourse relation classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 1006–1017.

Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1192–1202, Berlin, Germany, August.

Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 12–21.

Chen Shi, Shujie Liu, Shuo Ren, Shi Feng, Mu Li, Ming Zhou, Xu Sun, and Houfeng Wang. 2016. Knowledge-based semantic embedding for machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2245–2254, Berlin, Germany, August.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The conll 2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning - Shared Task (CoNLL)*, pages 159–177, Manchester, England, August.

Peilu Wang, Yao Qian, Frank K. Soong, Lei He, and Hai Zhao. 2016. Learning distributed word representations for bidirectional lstm recurrent neural network. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2016)*, pages 527–533.

Deyi Xiong, Min Zhang, and Haizhou Li. 2012. Modeling the translation of predicate-argument structure for smt. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 902–911, Jeju Island, Korea, July.

Zhuosheng Zhang and Hai Zhao. 2018. One-shot learning for question-answering in gaokao history challenge. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*.

Zhisong Zhang, Hai Zhao, and Lianhui Qin. 2016. Probabilistic graph-based dependency parsing with convolutional neural network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1382–1392.

Hai Zhao and Chunyu Kit. 2008. Parsing syntactic and semantic dependencies with two single-stage maximum entropy models. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning (CoNLL)*, pages 203–207.

Hai Zhao, Wenliang Chen, Jun'ichi Kazama, Kiyotaka Uchimoto, and Kentaro Torisawa. 2009a. Multilingual dependency learning: Exploiting rich features for tagging syntactic and semantic dependencies. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning - Shared Task (CoNLL)*, pages 61–66, Boulder, Colorado, June.

Hai Zhao, Wenliang Chen, and Chunyu Kit. 2009b. Semantic dependency parsing of NomBank and PropBank: An efficient integrated approach via a large-scale feature selection. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 30–39, Singapore, August.

Hai Zhao, Wenliang Chen, and Guodong Zhou. 2009c. Multilingual dependency learning: A huge feature engineering method to semantic dependency parsing. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning - Shared Task (CoNLL)*, pages 55–60, Boulder, Colorado, June.

Hai Zhao, Xiaotian Zhang, and Chunyu Kit. 2013. Integrative semantic dependency parsing via efficient large-scale feature selection. *Journal of Artificial Intelligence Research*, 46:203–233.

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 (ACL)*, pages 1127–1137, Beijing, China, July.