# A Unified Architecture for Semantic Role Labeling and Relation Classification

**Jiang Guo**[♪], **Wanxiang Che**[♪], **Haifeng Wang**[♩], **Ting Liu**[♪] and **Jun Xu**[♪]
[♪]Center for Social Computing and Information Retrieval, Harbin Institute of Technology, China
[♩]Baidu Inc., China
{jguo, car, tliu, jxu}@ir.hit.edu.cn
wanghaifeng@baidu.com

## Abstract

This paper describes a unified neural architecture for identifying and classifying multi-typed semantic relations between words in a sentence. We investigate two typical and well-studied tasks: *semantic role labeling* (SRL) which identifies the relations between predicates and arguments, and *relation classification* (RC) which focuses on the relation between two entities or nominals. While mostly studied separately in prior work, we show that the two tasks can be effectively connected and modeled using a general architecture. Experiments on CoNLL-2009 benchmark datasets show that our SRL models significantly outperform state-of-the-art approaches. Our RC models also yield competitive performance with the best published records. Furthermore, we show that the two tasks can be trained jointly with multi-task learning, resulting in additive significant improvements for SRL.

## 1 Introduction

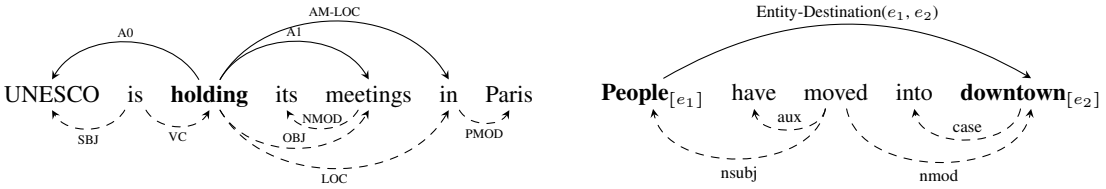
Semantic relation identification and classification are important problems towards the understanding of natural language sentences. Multi-typed semantic relations have been defined between two terms in a sentence in natural language processing (NLP) to promote various applications. For instance, the task of *Semantic Role Labeling* (SRL) defines shallow semantic dependencies between arguments and predicates, identifying the semantic roles, e.g., *who did what to whom, where, when, and how*. SRL has been a long-standing and challenging problem in NLP, primarily because it is strongly dependent on rich contextual and syntactical features used by the underlying classifiers (Gildea and Jurafsky, 2002). Another instance is *Relation Classification* (RC) which assigns sentences with two marked entities (or nominals) to a predefined set of relations (Hendrickx et al., 2010). Compared with SRL, relations defined in RC express much deeper semantics. Figure 1 shows example annotations of SRL and RC respectively.

These two problems are typically studied separately in different communities. Hence the connections between them are neglected, both in data resources and approaches. In this paper, we show that SRL and RC have a lot of common ground and can be modeled with a unified model. We start by looking into the key features which have been proven dominant in both SRL and RC.

- Contextual features. Words within a proper window size of the target words are important for most statistical models of various NLP tasks, such as Part-of-Speech tagging, Named Entity Recognition and Parsing. They are also important for identifying the semantic relatedness between two terms in a sentence. Consider the RC example in Figure 1(b), the context word "*moved*" is a strong indicator for classifying the relation of ⟨**People**, **downtown**⟩ as *Entity-Destination*. However, most of the conventional approaches in SRL and RC only considers local context features through feature engineering, which might be incomplete.

- Syntactical features. Both state-of-the-art SRL and RC systems employ the syntactic path between the two target terms as an important feature. Figure 1 shows the dependency parses for

† Corresponding author: Wanxiang Che




(a) Semantic Role Labeling. (b) Relation Classification.

Figure 1: Examples of *semantic role labeling* (a) and *relation classification* (b).

the two sentences. For example, the dependency path between "meetings" and the predicate "**holding**" ("holdings" $\xrightarrow{OBJ}$ "meetings") strongly indicates an A1 relation (*patient* role). Early approaches built in discrete feature space are not capable of utilizing the *word path* features which are extremely sparse. Fortunately, recent progress in distributed representations and deep neural networks provides a promising solution for this problem.

- Lexical semantic features. Lexical properties of a word (e.g., the identity of a word, its lemma, its morphological features) are important for semantic tasks. Particularly in tasks like relation classification, it is often impossible to determine the relation without the semantic ground of the target words. Therefore, previous approaches have been using lexical features like word embeddings, lemmas, WordNet, etc.

This paper describes a unified neural model for SRL and RC that effectively utilizes the three kinds of features above. Our model captures global contextual features and syntactic path features by using bidirectional long short-term memory (LSTM)-based recurrent neural networks. We especially focus on SRL which, in our opinion, is more complicated and difficult. SRL is a structure prediction task with certain structural constraints. To this end, an additional post-inference procedure based on integer linear programming (ILP) is applied to SRL, in order to meet the constraints. Furthermore, our unified model successfully connects SRL and RC, presenting the possibility of *multi-task learning*. We show that the SRL performance can be significantly improved through knowledge transfer from RC.

We conduct experiments on the CoNLL-2009 shared task datasets for SRL (Hajič et al., 2009), and the SemEval-2010 Task 8 dataset for RC (Hendrickx et al., 2010). On SRL, our models significantly outperform previous approaches in various languages. On RC, our model also obtains performance competitive to the state-of-the-art.[1]

Our primary original contributions include:

- We propose a unified model for SRL and RC, which effectively captures global contextual features, syntactical features and lexical semantic features.
- We show that SRL can be significantly improved by jointly training with RC, reaching new state-of-the-art performance.

## 2 Related Work

The present work ties together several strands of previous studies.

**Semantic Role Labeling** A great deal of previous SRL research has been dedicated to designing rich and expressive features, pioneered by Gildea and Jurafsky (2002). For instance, the top performing system on the CoNLL-2009 shared task employs over 50 language-specific feature templates (Che et al., 2009). These features mostly involve the predicate, the candidate argument, their contexts and the syntactic path between them (Surdeanu et al., 2003; Xue and Palmer, 2004; Pradhan et al., 2005). Besides, higher-order features involving several arguments or multiple predicates have also been explored (Toutanova et al., 2008; Martins and Almeida, 2014; Yang and Zong, 2014).

[1] Our code is available at: `https://github.com/jiangfeng1124/nnsrl-rc`.

Several approaches have been studied to alleviate the intensive feature engineering in SRL and get better generalization. Moschitti et al. (2008) introduce different kinds of tree kernels for capturing the structural similarity of syntactic trees. While attractive in automatic feature learning, the kernel-based approaches typically suffer from high computational cost. Lei et al. (2015) instead use low-rank tensors for automatic feature composition based on four kinds of basic feature sets. However, tensor-based approaches cannot well generalize the high-sparsity structural features like syntactic path. Besides, they still need a relatively small amount of feature engineering to make use of the local contexts. Another line of research focuses on neural models (Collobert et al., 2011; Zhou and Xu, 2015; FitzGerald et al., 2015), which have shown great effectiveness in automatic feature learning on a variety of NLP tasks. Most recently, Roth and Lapata (2016) employ LSTM-based recurrent neural networks to obtain the representations of syntactic path features, which is similar to our work. Aside from the distributed path features, they also use a set of binary input feature sets from Anders et al. (2010). In contrast to these prior work, our model jointly leverages both global contexts and syntactic path features using bidirectional LSTMs.

**Relation Classification** Early research on RC has also been relying heavily on human-engineered features (Rink and Harabagiu, 2010). Recent years have seen a great deal of work on using neural networks to alleviate the intensive engineering on contextual and syntactic features. For example, Socher et al. (2012) propose recursive neural networks for modeling the syntactic paths between the two entities whose relation is to be determined. Zeng et al. (2014) use convolutional neural network for learning sentence-level features of contexts and obtain good performance even without using syntactic features. Later approaches have used more sophisticated models for better handling long-term dependencies, such as sequential LSTMs and tree LSTMs (Liu et al., 2015; Xu et al., 2015b; Miwa and Bansal, 2016). In addition, Yu et al. (2014) and (2015) investigate tensor-based approaches for learning the combination of embedding features and lexicalized sparse features.

Therefore, despite that relation classification has mostly been studied separately from SRL, they have a substantial amount of commonalities. It inspires us to develop a potentially unified architecture to take advantage of the progress in each research direction.

**Multi-task Learning** There has been a line of research on joint modeling pipelined NLP tasks, such as word segmentation, POS tagging, parsing and semantic role labeling (Hatori et al., 2012; Li et al., 2011; Bohnet and Nivre, 2012; Henderson et al., 2013; Lluís et al., 2013). Most multi-task learning or joint training frameworks can be summarized as parameter sharing approaches proposed by Ando and Zhang (2005). In the context of neural modeling for NLP, the most notable work was proposed by Collobert and Weston (2008), which aims at solving multiple NLP tasks within one framework by sharing common word embeddings. This work also inspires us in this study to develop a unified architecture for SRL and RC in prior to joint training.

Recently, the idea of neural multi-task learning was applied to sequence-to-sequence problems with recurrent neural networks. Dong et al. (2015) use multiple decoders in neural machine translation systems that allows translating one source language to many target languages. Luong et al. (2015) study the ensemble of a wide range of tasks (e.g., syntactic parsing, machine translation, image caption, etc.) with multi-task sequence-to-sequence models. Liu et al. (2016) incorporate different kinds of corpus for implicit discourse relation classification using multi-task neural networks. More recently, multi-task learning has also been applied to sentence compression (Klerke et al., 2016) and machine translation quality estimation (Shah and Specia, 2016).

## 3 Problem Definition

This section gives formal definitions of the two tasks to be investigated: SRL and RC.

### 3.1 Semantic Role Labeling

We follow the setup of the CoNLL-2009 shared task. Given a sentence $\mathbf{s}$, each token is annotated with a predicated POS tag and predicted word lemma. Some tokens are also marked as predicates. Besides,

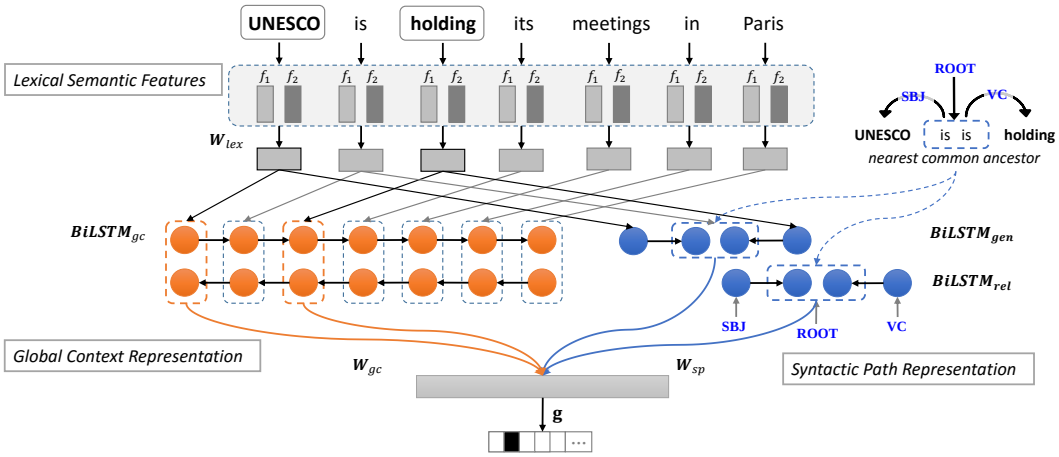



Figure 2: The unified architecture for SRL and RC.

a predicted syntactic dependency tree $y_{syn}$ is also provided (cf. below part of Figure 1(a)). The goal is to determine the semantic dependencies for each predicate $p_i$ (cf. upper part of Figure 1(a)). These dependencies identify the arguments of each predicate and the role labels.

In this work, we focus on the identification and classification of the arguments associated with given predicates. More formally, for each predicate $p_i$ in $\mathbf{s}$, we loop over all the tokens in $\mathbf{s}$ except $p_i$: $\{w \in \mathbf{s} | w \neq p_i\}$, and determine their role labels. It can be considered as a classification problem with each instance as a word pair $\langle p_i, w \rangle$. We include an additional *NULL* label indicating that a token is not an argument of $p_i$. To guarantee the resulting semantic dependencies meet certain constraints, we further apply ILP over the output probabilities in each position for post-inference (Section 4.4).

### 3.2 Relation Classification

As demonstrated in Figure 1, the semantic relations specified in relation classification are totally different from SRL. SRL is more close to the syntactic dependencies while RC is totally semantic. Our setup follows the SemEval-2010 Task 8. Each sentence $\mathbf{s}$ is annotated with a pair of nominals $e_1$ and $e_2$, and our goal is to identify the relation between $e_1$ and $e_2$. Nine relations are defined in the task, and the directionality of relation between $e_1$ and $e_2$ is considered in the evaluation. Relations that do not belong to the nine relations are marked as *Other*.

## 4 Unified Neural Architecture

As described above, both SRL and RC can be formalized as a classification problem over instances of word pairs within a sentence. We propose a unified neural architecture, as illustrated in Figure 2, for modeling these two tasks. Our architecture includes the following three primary components.

### 4.1 Lexical Feature Representation

We extract basic lexical features for each token in a sentence. Typical lexical features for SRL and RC include word (or *lemma* when available) and POS tag. For RC, additional features can be used, such as *named entity type* (NE) and WordNet. All these features are then represented as low-dimensional real-valued vectors, i.e., feature embeddings. Word embeddings can be readily pretrained using *word2vec* on a large unlabeled corpus, which have proved helpful in many applications. Next, various feature embeddings are composed through a nonlinear transformation, and thus a token can be represented as:

$$\begin{aligned} &\mathbf{x}_i = \texttt{ReLU}(\mathbf{W}_{lex}\mathbf{\Phi}_i + \mathbf{b}_{lex}), where \\ &\mathbf{\Phi}_i = [\mathbf{w}_i; \mathbf{p}_i] \ \ for \ \ \textbf{SRL}, \ \ \mathbf{\Phi}_i = [\mathbf{w}_i; \mathbf{p}_i; \mathbf{ne}_i; \mathbf{wn}_i] \ \ for \ \ \textbf{RC} \end{aligned} \tag{1}$$

$\mathbf{w}_i$ represents the word or lemma (when available), $\mathbf{p}_i$ represents the POS tag, $\mathbf{ne}_i$ is the named entity, and $\mathbf{wn}_i$ is the WordNet hypernym.

### 4.2 Global Context Representation

We obtain global context representations of the target words by using bidirectional LSTM-based RNNs. For more computation details of LSTM, we refer the readers to Hochreiter and Schmidhuber (1997). The LSTMs take as input the token representation $\mathbf{x}_i$ in each position. The hidden state vectors of the two directions' LSTM units corresponding to each target word are then concatenated as its global context representation:

$$\mathbf{R}^{gc}_{e_1} = [\overrightarrow{\mathbf{h}}_{e_1}; \overleftarrow{\mathbf{h}}_{e_1}]; \quad \mathbf{R}^{gc}_{e_2} = [\overrightarrow{\mathbf{h}}_{e_2}; \overleftarrow{\mathbf{h}}_{e_2}] \tag{2}$$

Note that an important difference between our model and previous neural models is that we utilize the hidden state vectors of $e_1$ and $e_2$ instead of the representation of the whole sentence, which frees us from using position-related features (Zeng et al., 2014; Collobert et al., 2011; dos Santos et al., 2015).

### 4.3 Syntactic Path Representation

We define the *nearest common ancestor* token of $e_1$ and $e_2$ as $nca(e_1, e_2)$. Then the path from $e_1$, $e_2$ to $nca(e_1, e_2)$, i.e., $e_1 \rightarrow \ldots \rightarrow nca(e_1, e_2)$ and $nca(e_1, e_2) \leftarrow \ldots \leftarrow e_2$, are also modeled with bidirectional LSTMs, as shown in Figure 2 (right panel). We use two kinds of syntactic paths, including a *generic path* that takes the token representation $\mathbf{x}_i$ as input, and a *relation path* that takes the dependency relations along the path as input (Figure 2). These two paths are modeled with $\mathbf{BiLSTM}_{gen}$ and $\mathbf{BiLSTM}_{rel}$ respectively. The hidden state vectors of the two directions' LSTM units of $nca(e_1, e_2)$ are then concatenated as the syntactic path representation of $(e_1, e_2)$:

$$\mathbf{R}^{gen}_{(e_1,e_2)} = [\overrightarrow{\mathbf{h}}^{gen}_{nca(e_1,e_2)}; \overleftarrow{\mathbf{h}}^{gen}_{nca(e_1,e_2)}]; \quad \mathbf{R}^{rel}_{(e_1,e_2)} = [\overrightarrow{\mathbf{h}}^{rel}_{nca(e_1,e_2)}; \overleftarrow{\mathbf{h}}^{rel}_{nca(e_1,e_2)}] \tag{3}$$

The global context representations and syntactic path representation are then composed through a non-linear layer, resulting in the representation used for final classification.

$$\mathbf{p} = \texttt{ReLU}(\mathbf{W}_{gc} \underbrace{[\mathbf{R}^{gc}_{e_1}; \ \mathbf{R}^{gc}_{e_2}]}_{\text{Global Context}} + \mathbf{W}_{sp} \underbrace{[\mathbf{R}^{gen}_{(e_1,e_2)}; \ \mathbf{R}^{rel}_{(e_1,e_2)}]}_{\text{Syntactic Path}} + \mathbf{b}) \tag{4}$$

$$p(c|\mathbf{p}) = \texttt{softmax}(\mathbf{g}_c^\top \mathbf{p} + \mathbf{q}_c) \tag{5}$$

Our model is trained by minimizing the cross-entropy loss: $\mathcal{L}(\theta) = -\sum_{i=0}^{N} \log p(c_i|\mathbf{p}_i)$, where $N$ is number of training instances.

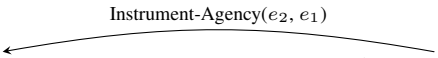
### 4.4 Post-Inference with Integer Linear Programming for SRL

SRL is a structure prediction problem and the predicted results should satisfy some structural constraints. For instance, some roles only appear once for a predicate in a sentence. Following Punyakanok et al. (2004) and Che et al. (2008), we apply ILP on the probability distributions at each token generated by our model to get the global optimization. We use the three constraints defined in Che et al. (2008):

- C1: Each word should be labeled with one and only one label (including *NULL*).
- C2: Roles with a small probability (smaller than 0.3) should never be labeled (except for *NULL*).
- C3: Some roles (except for *NULL*) usually appear once for a predicate in a sentence. Hence a non-duplicate-roles list is utilized for each language.

## 5 Multi-task Learning

The commonalities between SRL and RC inspire us to explore their potential mutual benefits. According to the *Shortest Path Hypothesis* (Bunescu and Mooney, 2005), if $e_1$ and $e_2$ are two entities mentioned in the same sentence such that they are observed to be in a certain relationship $R$, they often indicate two arguments of the same predicate or a sequence of predicates. To gain more insights, let's look at the following example in RC:




"The **author**$_{[e_1]}$ of a keygen uses a **disassembler**$_{[e_2]}$ to look at the raw assembly code."

Here, the "Instrument-Agency" relation provides significant evidences that **author** and **disassembler** are two arguments of a certain predicate, most likely with semantic roles A0 (agent) and A1 (patient). Furthermore, given the dependency parse tree, it's easy to find out that their associated predicate is "uses". Therefore, RC is expected to benefit both the identification and classification of semantic roles. Analogously, SRL results of a sentence also have positive impacts to the *identification* of semantic relations between $e_1$ and $e_2$, i.e. whether or not a relation exists between $e_1$ and $e_2$.

However, the roles defined in SRL can hardly contribute to the *classification* of much more fine-grained relation types in RC. For example, the roles A0, A1 can hardly help us to distinguish between the relation types like *Instrument-Agency*, *Product-Producer*, *Cause-Effect*, etc. Given this intuition, we will mainly focus on improving SRL with RC in this work.

Our proposed unified model allows knowledge transfer across SRL and RC in a natural way through parameter sharing. In this work, we consider two ways of knowledge transfer.

- Cascaded Learning (CAS). Models are trained in a cascaded manner. Specifically, a RC model is trained first, and then the parameters (e.g., word embeddings, network weights) are used to initialize the neural network for training SRL in the second stage.
- Multi-task Learning (MTL). Models are trained jointly in a stochastic manner:
  1. Select a task according to a certain probability distribution (explained below).
  2. Sample a batch of instances from the task, and feed-forward the neural network.
  3. Update the corresponding parameters by back-propagation w.r.t. the instances.
  4. Go to 1.

In multi-task learning, two important factors are taken into account. First, we typically expect the two tasks to converge at a similar rate (Caruana, 1997). We approximately achieve this by using a **weighted task sampling** strategy in step 1. More specifically, we observe that SRL converges about 4 times slower than RC by running them separately, hence we sample from SRL 4 times often than RC during training. Despite the lack of theoretical guarantee, we found it working well in practice. Second, the key for multi-task learning to work is **parameter sharing**. Given the unified architecture, we can share most of the network parameters for knowledge transfer. Note that different dependency parses might be used for SRL and RC in practice. In this work, we use the officially provided predicted parses from CoNLL-2009 shared task in SRL, but adopt Stanford parser (Manning et al., 2014) to obtain parses for sentences in RC. These kinds of parses are quite different in terms of both the head-finding rules and the dependency relations. Therefore, we set the parameters involving dependency path modeling as *task-specific*, i.e., $\mathbf{BiLSTM}_{gen}, \mathbf{BiLSTM}_{rel}$ and $\mathbf{W}_{sp}$ (Figure 2). The output weights ($\mathbf{g}$) are *task-specific* as standard of multi-task learning, in order to handle different set of relations to be classified in SRL and RC.

## 6 Experiment

In this section, we first describe data and our experimental settings, then the results and analysis.

### 6.1 Data and Settings

For SRL, we evaluate on the English dataset and other 4 languages (Chinese, Catalan, German and Spanish) in the CoNLL-2009 shared task. We use the official split for training, development and testing. In addition, a subset of the Brown corpus is used as the out-of-domain test set. We use the officially provided predicted POS tags, lemmas and dependency parses as our input. All predicates are given for each sentence during both training and testing. Besides, we neither predict nor use the sense for each predicate, and thus exclude the predicate senses in most of the evaluation. We follow Lei et al. (2015) and combine the predicate sense output of Anders et al. (2010) with our SRL output, to provide results directly comparable to previous published results.

We compare our model to several state-of-the-art systems, primarily including the best performing system in CoNLL-2009 shared task, the most recently proposed PathLSTM model of Roth and Lapata (2016), the neural network model of FitzGerald et al. (2015), and the low-rank tensor model of Lei et al. (2015). We also consider some variants of the above models that use reranking or model ensemble.

For RC, we use the relation classification dataset of the SemEval 2010 task 8. The dataset contains 10,717 annotated sentences, including 8,000 for training and 2,717 for testing. We conduct 5-fold cross-validation to determine the best training iterations, and use the official scoring script for evaluation.

Several competitive models are to be compared, including the top performed system in SemEval 2010 (Rink and Harabagiu, 2010), the Matrix-Vector Recursive Neural Network (MV-RNN) model of Socher et al. (2012), the CNN model of Zeng et al. (2014), the tensor-based model of Yu et al. (2014), the CNN model using ranking loss (dos Santos et al., 2015), and the dependency-based neural network models (Liu et al., 2015; Xu et al., 2015b).

Word embeddings are pretrained using *word2vec* on large-scale unlabeled data. For English, Catalan, German and Spanish, we use the latest Wikipedia data. For Chinese, we obtain the raw text from Xinhua news section (2000–2010) of the fifth edition of Chinese Gigaword (LDC2011T13). The LTP toolkit (Che et al., 2010) is applied to segment Chinese text into words.

We adopt *predicate-wise* training for SRL and *sentence-wise* training for RC, and use stochastic gradient descent for optimization. Initial learning rate is set to $\eta_0 = 0.1$ and updated as $\eta_t = \eta_0/(1+0.1t)$ on each epoch $t$. Our hyperparameters for the unified model are listed in Table 1. When training RC-only models, the LSTM input/hidden dimension is set to 200, and the dimension of hidden layer is 400.

| Dimension of embeddings | | | | Dimension of layers | | |
|---|---|---|---|---|---|---|
| *word* | *POS* | *NE* | *WordNet* | *LSTM input* | *LSTM hidden* | *hidden* |
| 200 | 25 | 25 | 25 | 100 | 100 | 200 |

Table 1: Hyperparameters settings.

### 6.2 SRL Results

Table 2 reports the SRL performance on the English dataset. Our supervised models (SUP) outperform the six top performing systems on both in-domain and out-of-domain datasets (the second block), and is comparable to two top systems that use reranking or model ensemble (the third block).

**Effect of transfer learning** By comparing the cascaded training system (CAS), the multi-task learning system (MTL) with SUP, we can find that the task of RC is significantly helpful for improving SRL models. In particular, MTL consistently works better than CAS. Our best models (MTL) outperform all of the previous systems, and achieve new state-of-the-art SRL results.

Figure 3 shows the learning curves of SUP, CAS and MTL on development data. At early training iterations, CAS is very close to MTL, and improves faster than SUP, indicating that the RC parameters indeed serve as a good initialization for SRL. MTL gradually outperforms CAS as the training converges, which further verifies the advantage of joint training over cascaded training.

**Effect of post-inference** We further investigate the effect of post-inference with ILP. As shown in Table 3, ILP has a considerable impact on the final SRL performance consistently for all of our models.

**Multilingual Results** Table 4 shows the results of our SRL-only system (SUP) on other languages in the CoNLL-2009 shared task. Our model outperforms the best performing system on all the four languages we considered, with particularly large gains on Chinese (+6.3 absolute F1-score). Note that our model is also unified for each language, without language-specific tuning of features or hyperparameters.

### 6.3 RC Results

The only difference of our RC model from the SRL model is at the input layer, where we use two additional features: NE and WordNet. Table 5 shows the RC results on the SemEval 2010 task 8. Our model achieves an F1-score of 83.9%, which is comparable to the top performing systems in previous work. dos Santos et al. (2015) obtain an F1-score of 84.1% by using ranking loss, with special treatment

| Model | Excluding predicate senses | | | Including predicate senses | |
|---|---|---|---|---|---|
| | WSJ-dev | WSJ-test | Brown-test | WSJ-test | Brown-test |
| SUP | 82.32 | 84.06 | 72.12 | 87.67 | 76.56 |
| CAS | 83.33 | 84.73 | 73.00 | 88.14 | 77.15 |
| MTL | **83.51*** | **85.04*** | **73.22*** | **88.37*** | **77.34*** |
| CoNLL-2009 1st place | – | 82.08 | 69.84 | 86.15 | 74.58 |
| (Roth and Lapata, 2016) | – | – | – | 86.7 | 75.3 |
| (FitzGerald et al., 2015) | 82.3 | 83.6 | 71.9 | 87.3 | 75.2 |
| (Lei et al., 2015) | 81.03 | 82.51 | 70.77 | 86.58 | 75.57 |
| (Roth and Woodsend, 2014) | – | 80.87 | 69.33 | 85.50 | 74.67 |
| (Anders et al., 2010) | 78.85 | 81.35 | 68.34 | 85.80 | 73.92 |
| **Model + Reranker/Ensemble** | WSJ-dev | WSJ-test | Brown-test | WSJ-test | Brown-test |
| (Roth and Lapata, 2016)+R,E | – | – | – | 87.9 | 76.5 |
| (FitzGerald et al., 2015)+E | 83.0 | 84.3 | 72.4 | 87.8 | 75.5 |
| (Roth and Woodsend, 2014)+R | – | 82.10 | 71.12 | 86.34 | 75.88 |
| (Anders et al., 2010)+R | 80.50 | 82.87 | 70.91 | 86.86 | 75.71 |

Table 2: SRL labeled F1-score of our model variants, with comparison to the state-of-the-art systems on the CoNLL-2009 shared task. Statistical significance (MTL *vs.* SUP) with $p < 0.01$ is marked with *.

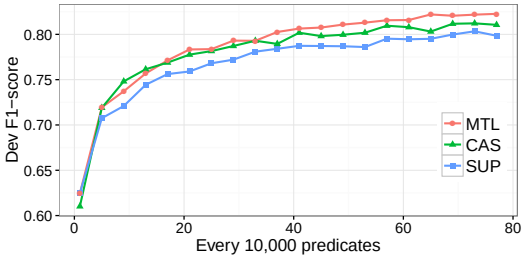



Figure 3: SRL F1-scores on the development data w.r.t. the number of predicates trained.

| Model | WSJ-dev | WSJ-test |
|---|---|---|
| SUP | 82.32 | 84.06 |
| w/o ILP | 81.87 | 83.53 |
| CAS | 83.33 | 84.73 |
| w/o ILP | 82.90 | 84.40 |
| MTL | 83.51 | 85.04 |
| w/o ILP | 83.15 | 84.75 |

Table 3: Effect of post-inference, evaluated excluding predicate senses.

to the artificial relation (*Other*). Such task-specific strategy can also be potentially used in our model for further improvements. As discussed in Section 5, to our intuition, knowledge contained in SRL is not supposed to benefit RC. To verify this, we further test on RC with cascaded learning and multi-task learning. We obtain a small degradation in RC performance in both cases (-0.9 for CAS and -0.7 for MTL). Nevertheless, we still expect improvements on joint learning of SRL and *relation extraction* (rather than *classification*), which we leave to future exploration.

## 7 Conclusion

In this paper, we propose a unified architecture for the task of SRL and RC. We effectively capture the global contextual representation and syntactic path representations using bidirectional LSTM-based recurrent neural networks. By evaluating on benchmark datasets for both SRL and RC, we show that our models outperform or get competitive results with the state-of-the-art systems. Furthermore, we take advantage of our unified model to transfer knowledge across the two tasks using multi-task learning with parameter sharing. Our models obtain new state-of-the-art results for SRL.

## Acknowledgements

We are grateful to Tao Lei for providing the outputs of their systems. We thank the anonymous reviewers for their insightful comments and suggestions. This work was supported by the National Key Basic Research Program of China via grant 2014CB340503 and the National Natural Science Foundation of China (NSFC) via grant 61300113 and 61370164.

| Language | Test set | | | |
|---|---|---|---|---|
| | Ours | (Lei et al., 2015) | CoNLL 1st | CoNLL 2nd |
| Chinese | **75.46** | 69.16 | 68.52 | 68.71 |
| Catalan | **79.24** | 74.67 | 76.78 | 74.02 |
| German | **77.41** | 76.94 | 74.65 | 76.27 |
| Spanish | **79.17** | 75.58 | 77.33 | 74.01 |

Table 4: SRL labeled F1-score excluding predicate senses on Chinese, Catalan, German and Spanish. All results are evaluated excluding predicate senses.

| **Model** | **Features** | **F1** |
|---|---|---|
| SVM (Rink and Harabagiu, 2010) (Best in SemEval 2010) | POS, prefixes, morphological, WordNet, Levin classes, PropBank, FrameNet, dependency parse, NomLex-Plus, Google n-gram, paraphrases, TextRunner | 82.2 |
| MVRNN (Socher et al., 2012) | syntactic parse | 79.1 |
| MVRNN (Socher et al., 2012) | syntactic parse, POS, NER, WordNet | 82.4 |
| CNN (Zeng et al., 2014) | position, WordNet | 82.7 |
| FCM (Yu et al., 2014) | dependency path, NER | 83.0 |
| DepNN (Liu et al., 2015) | dependency parse, NER | 83.6 |
| CR-CNN (dos Santos et al., 2015) | position | **84.1** |
| depLCNN (Xu et al., 2015a) | WordNet, words around nominals | 83.7 |
| Ours | dependency path, POS, NER, WordNet | **83.9** |
| **Model + Ensemble/Additional data** | | |
| ER-CNN+R-RNN (Vu et al., 2016) | position | 84.9 |
| depLCNN+NS (Xu et al., 2015a) | WordNet, words around nominals | 85.6 |

Table 5: Comparison with previously published results for SemEval 2010 Task 8.

## References


Björkelund Anders, Bohnet Bernd, Love Hafdell, and Pierre Nugues. 2010. A high-performance syntactic and semantic dependency parser. In *Coling 2010: Demonstrations*, pages 33–36, August.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR*, 6:1817–1853, December.

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proc. of EMNLP-CoNLL*, pages 1455–1465, July.

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proc. of EMNLP*, pages 724–731, October.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Wanxiang Che, Zhenghua Li, Yuxuan Hu, Yongqiang Li, Bing Qin, Ting Liu, and Sheng Li. 2008. A cascaded syntactic and semantic dependency parsing system. In *Proc. of CoNLL*, pages 238–242, August.

Wanxiang Che, Zhenghua Li, Yongqiang Li, Yuhang Guo, Bing Qin, and Ting Liu. 2009. Multilingual dependency-based syntactic and semantic parsing. In *Proc. of CoNLL 2009: Shared Task*, pages 49–54, June.

Wanxiang Che, Zhenghua Li, and Ting Liu. 2010. Ltp: A chinese language technology platform. In *Coling 2010: Demonstrations*, pages 13–16, August.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. of the 25th ICML*, pages 160–167.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proc. of the 53rd ACL and the 7th IJCNLP (Volume 1: Long Papers)*, pages 1723–1732, July.

Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proc. of ACL-IJCNLP*, pages 626–634, July.

Nicholas FitzGerald, Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Semantic role labeling with neural network factors. In *Proc. of EMNLP*, pages 960–970, September.

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

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 *Proc. of CoNLL 2009: Shared Task*, pages 1–18, June.

Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2012. Incremental joint approach to word segmentation, pos tagging, and dependency parsing in chinese. In *Proc. of ACL*, pages 1045–1053, July.

James Henderson, Paola Merlo, Ivan Titov, and Gabriele Musillo. 2013. Multilingual joint parsing of syntactic and semantic dependencies with a latent variable model. *Computational Linguistics*, 39(4):949–998.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proc. of SemEval*, pages 33–38, July.

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

Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016. Improving sentence compression by learning to predict gaze. In *Proc. of NAACL*, pages 1528–1533, June.

Tao Lei, Yuan Zhang, Lluís Màrquez, Alessandro Moschitti, and Regina Barzilay. 2015. High-order low-rank tensors for semantic role labeling. In *Proc. of NAACL*, pages 1150–1160, May–June.

Zhenghua Li, Min Zhang, Wanxiang Che, Ting Liu, Wenliang Chen, and Haizhou Li. 2011. Joint models for chinese pos tagging and dependency parsing. In *Proc. of EMNLP*, pages 1180–1191, July.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng WANG. 2015. A dependency-based neural network for relation classification. In *Proc. of ACL-IJCNLP*, pages 285–290, July.

Yang Liu, Sujian Li, Xiaodong Zhang, and Zhifang Sui. 2016. Implicit discourse relation classification via multi-task neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Xavier Lluís, Xavier Carreras, and Lluís Màrquez. 2013. Joint arc-factored parsing of syntactic and semantic dependencies. *Transactions of the Association for Computational Linguistics*, 1:219–230.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proc. of 52nd ACL: System Demonstrations*, pages 55–60, June.

André F. T. Martins and Mariana S. C. Almeida. 2014. Priberam: A turbo semantic parser with second order features. In *Proc. of SemEval 2014*, pages 471–476, August.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*.

Alessandro Moschitti, Daniele Pighin, and Roberto Basili. 2008. Tree kernels for semantic role labeling. *Computational Linguistics*, 34(2):193–224.

Sameer Pradhan, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2005. Support vector learning for semantic argument classification. *Machine Learning*, 60(1-3):11–39, September.

Vasin Punyakanok, Dan Roth, Wen-tau Yih, and Dav Zimak. 2004. Semantic role labeling via integer linear programming inference. In *Proc. of COLING*, pages 1346–1352, Aug 23–Aug 27.

Bryan Rink and Sanda Harabagiu. 2010. Utd: Classifying semantic relations by combining lexical and semantic resources. In *Proc. of the 5th International Workshop on Semantic Evaluation*, pages 256–259, July.

Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. *arXiv preprint arXiv:1605.07515*.

Michael Roth and Kristian Woodsend. 2014. Composition of word representations improves semantic role labelling. In *Proc. of EMNLP*, pages 407–413, October.

Kashif Shah and Lucia Specia. 2016. Large-scale multitask learning for machine translation quality estimation. In *Proc. of NAACL*, pages 558–567, June.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proc. of EMNLP-CoNLL*, pages 1201–1211, July.

Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using predicate-argument structures for information extraction. In *Proc. of ACL*, pages 8–15, July.

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. 2008. A global joint model for semantic role labeling. *Computational Linguistics*, 34(2):161–191, June.

Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016. Combining recurrent and convolutional neural networks for relation classification. In *Proc. of NAACL*, pages 534–539, June.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015a. Semantic relation classification via convolutional neural networks with simple negative sampling. In *Proc. of EMNLP*, pages 536–540, September.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015b. Classifying relations via long short term memory networks along shortest dependency paths. In *Proc. of EMNLP*, pages 1785–1794, September.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *Proc. of EMNLP*, pages 88–94, July.

Haitong Yang and Chengqing Zong. 2014. Multi-predicate semantic role labeling. In *Proc. of EMNLP*, pages 363–373, October.

Mo Yu, Matthew Gormley, and Mark Dredze. 2014. Factor-based compositional embedding models. In *NIPS Workshop on Learning Semantics*.

Mo Yu, Matthew R. Gormley, and Mark Dredze. 2015. Combining word embeddings and feature embeddings for fine-grained relation extraction. In *Proc. of NAACL*, pages 1374–1379, May–June.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proc. of COLING*, pages 2335–2344, August.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proc. of ACL-IJCNLP*, pages 1127–1137, July.