

Jointly Part-of-Speech Tagging and Semantic Role Labeling Using Auxiliary Deep Neural Network Model

Yatian Shen¹, Yubo Mai², Xiajiong Shen², Wenke Ding^{2,*} and Mengjiao Guo³

Abstract: Previous studies have shown that there is potential semantic dependency between part-of-speech and semantic roles. At the same time, the predicate-argument structure in a sentence is important information for semantic role labeling task. In this work, we introduce the auxiliary deep neural network model, which models semantic dependency between part-of-speech and semantic roles and incorporates the information of predicate-argument into semantic role labeling. Based on the framework of joint learning, part-of-speech tagging is used as an auxiliary task to improve the result of the semantic role labeling. In addition, we introduce the argument recognition layer in the training process of the main task-semantic role labeling, so the argument-related structural information selected by the predicate through the attention mechanism is used to assist the main task. Because the model makes full use of the semantic dependency between part-of-speech and semantic roles and the structural information of predicate-argument, our model achieved the F1 value of 89.0% on the WSJ test set of CoNLL2005, which is superior to existing state-of-the-art model about 0.8%.

Keywords: Part-of-speech tagging, semantic role labeling, multi-task learning.

1 Introduction

Semantic role labeling (SRL) [Gildea and Jurafsky (2002)] is to assign appropriate semantic role to components or phrases in a sentence to answer the question: who did what to whom, where and when? This task is usually divided into two steps: the first step is to recognize predicates; the second step is to identify arguments and label semantic roles. SRL has been standardized in different frameworks, the most prominent being FrameNet and PropBank [Palmer, Babko-Malaya, Bies et al. (2008)]. As a basic and important task in natural language processing, SRL can provide rich semantic information for subsequent downstream tasks, such as Relation Extraction [Yin, Meng, Li et al. (2019)], Question Answering System [Zhang, Zhao and Qin (2016)] and Machine Translation [Shi, Liu, Ren et al. (2016)].

¹ School of Computer Science and Engineering, Southeast University, Nanjing, 210000, China.

² Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng, 475000, China.

³ Swinburne Data Science Research Institute, Swinburne University of Technology, Victoria, 3122, Australia.

* Corresponding Author: Wenke Ding. Email: xd@henu.edu.cn.

Received: 22 April 2020; Accepted: 20 May 2020.

Traditional methods mentioned above are all based on feature engineering, which take advantage of a large body of linguistic knowledge. However, these methods may suffer from some limitations. The method of artificial feature extraction is laborious and time-consuming, and these features do not scale well for other tasks. The latest end-to-end neural network models greatly improved the results for SRL in English typically [He, Lee, Lewis et al. (2017); Bastings, Titov, Aziz et al. (2017)]. In general, such models used a deep LSTM architecture to treat this problem as supervised sequence labeling task, which assigned a label to each token in the sentence.

However, supervised sequence labeling methods may suffer two limitations. The size of SRL corpus is very limited in many languages. Since a specific predicate-role instance only appears a few times in the training set, the model will encounter the problem of data sparsity. In addition, a mass of annotated SRL data is expensive.

To solve this problem, we make use of multi-task learning method for jointly accomplishing part-of-speech tagging and semantic role labeling, which takes part-of-speech tagging as the auxiliary task. Multi-task learning takes advantage of the correlation between similar tasks and improves the effect by learning parallel tasks [Collobert and Weston (2008)]. The basic architecture of these models is to use some lower layers to share features and parameters. Among them, the shared layers at the bottom are used for feature encoding, and the specific layers at the top are designed for different tasks.

In the task of semantic role labeling, the predicate is not only important, but also the relation between predicate and argument. But some models ignored the potential semantic relation between predicate and argument in previous work. So how to acquire the structure of predicate-argument is important, which can promote the result of semantic role labeling. We use multi-layer stacked Bi-LSTM as shared encoding layer of model, and part-of-speech tagging is considered as the auxiliary task to train alternately with semantic role labeling. The important thing is that the training times of the main and auxiliary tasks are different. In the training process of semantic role labeling, the predicate selects the information related to arguments through the attention mechanism, which is taken as part of semantic features. For the convenience of writing, we named our auxiliary deep neural network model as ANNM.

We evaluate our model on the benchmark of CoNLL2005. The results show that multi-task learning of part-of-speech tagging and semantic role labeling jointly can produce the best performance. At the same time, predicates can also choose important information related to the arguments. The F1 value of 89.0% is significantly superior to the best baseline about 0.8%.

In summary, our major contributions include: 1) First, part-of-speech tagging and semantic role labeling are unified into one model by using multi-task learning framework, and in the process of training, the main task and the auxiliary task are trained alternately. 2) During the training process of the main task, we add an argument recognition layer to the model, which gains the selection of argument-related information by predicates with the help of the attention mechanism. 3) Our model obtains the F1 value of 89.0% on the WSJ test set of CoNLL2005. Compared with the existing best model, the F1 value of semantic role labeling is increased by 0.8%.

2 Related work

Traditional semantic role labeling depended on the extraction of syntactic features and constraint rules. Punyakanok et al. [Punyakanok, Roth and Yih (2008); FitzGerald, Täckström, Ganchev et al. (2015)] respectively used integer linear programming and dynamic programming to achieve global consistency. Many researchers have tried to add syntactic features to neural network. Idrees et al. [Idrees and FitzGerald (2015)] used feed-forward neural networks to implement role representations and used graph models for global constraints. Roth et al. [Roth and Lapata (2016)] treated nested subordinations and nominal predicates as subsequences of dependency paths and learned corresponding embedding representations, and experiments showed that such embeddings could improve the performance of semantic role labeling model.

With the recent development in deep neural networks, some researchers have concentrated on using neural networks to semantic role labeling. Collobert et al. [Collobert, Weston, Bottou et al. (2011)] first used neural networks model without artificial features and viewed this task as sequence labeling task. Later, Zhou et al. [Zhou and Xu (2015)] proposed a deep Bi-directional LSTM model based on CRF layer. He et al. [He, Lee, Lewis et al. (2017)] utilized a highway Bi-LSTM structure with constrained decoding, which reduced the relative error by 10% compared with the previous best results. Marcheggiani et al. [Marcheggiani and Titov (2017)] combined the LSTM model with a graph convolutional network to encode syntactic information at word level. He et al. [He, Lee, Levy et al. (2018)] proposed an end-to-end method for joint prediction of all predicates and arguments spans, which overcame a key limitation of semi-Markov and BIO tagging. Cai et al. [Cai, He, Li et al. (2018)] made use of an end-to-end neural network that uniformly processed predicate disambiguation. Ouchi et al. [Ouchi, Shindo and Matsumoto (2018)] proposed an effective and simple span-based semantic role labeling model. And the model directly considered all possible argument spans and scored each label. Swayamdipta et al. [Swayamdipta, Thomson, Lee et al. (2018)] took advantage of the syntactic scaffold, which was an effective method to combine syntactic information into SRL tasks. Wang et al. [Wang, Johnson, Wan et al. (2019)] injected three different syntactic parses into a neural ELMo-based SRL model and evaluated their performance, to explore how to make efficient use of external syntactic information in semantic role labeling. Pal et al. [Pal and Sharma (2019)] collected a set of 1460 tweets mixed in Hindi and English, and created a verb framework for complex predicates in the corpus.

Different from the above work, ANNM proposed by us mainly uses the multi-task learning framework. The model extracts the sharing information through the joint learning of part-of-speech tagging and semantic role labeling, and completes the selection of argument-related information by predicates with the help of the attention mechanism.

3 Model

In this chapter, we first use word embedding layer in Section 3.1. Next, we briefly introduce the basic network of the model-Bi-LSTM in Section 3.2. Then we represent the argument recognition layer in Section 3.3 that can select the structure information of predicate-argument. The basic architecture of ANNM is shown in Section 3.4. Eventually, we describe the training process and loss function in Section 3.5 of the model

in detail. And the architecture of proposed model is shown in Fig. 1.

3.1 Embedding layer

In the word embedding stage, each word is initialized as a distributed vector. Collobert et al. [Collobert, Weston, Bottou et al. (2011)] found that word embedding trained from large amounts of unlabeled data could encode word information better than randomly initialized embedding. Some researchers trained a significant amount of unlabeled data to obtain word embedding which contained more syntactic and semantic information. However, learning word embedding from a mass of unlabeled data requires a large amount of time and hardware environment. Mikolov et al. [Mikolov, Chen, Corrado et al. (2013)] trained 100 billion Google news words to get excellent word embedding. And the word embedding is freely available. In our experiment, we also use this word vector mentioned above.

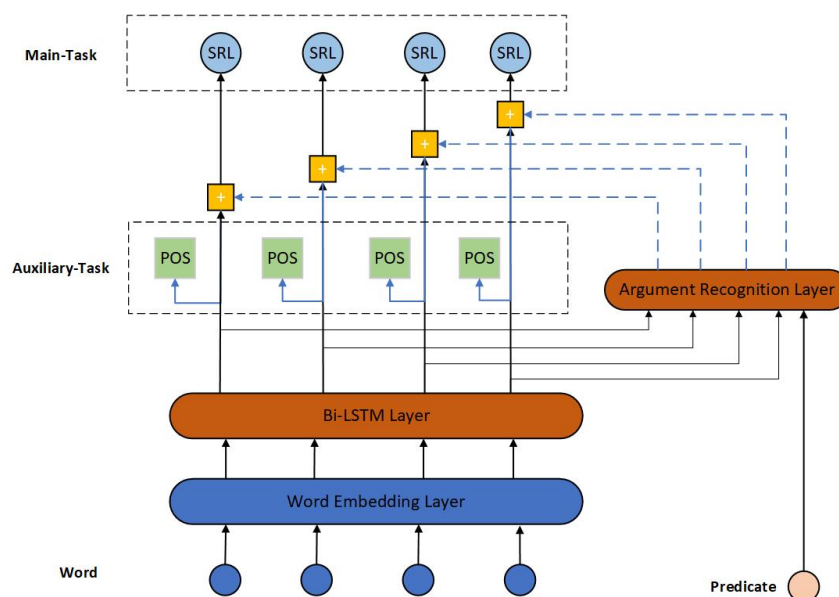


Figure 1: The architecture of the ANNM

3.2 Sentence encoding with Bi-LSTM

Recurrent neural network is the most common and effective representation method for processing sequence data. The input of the hidden layer includes not only the input of the current time step, but also the output of the hidden layer of the previous time step. The hidden layer of the recurrent neural network can maintain the intermediate state, but RNN also has the problems of gradient diffusion and gradient explosion. Cho et al. [Cho, Van Merriënboer, Gulcehre et al. (2014)] respectively proposed the improved models of recurrent neural network, namely Long Short-term Memory Unit (LSTM) and Gated Recurrent Unit (GRU). Two models introduced gate mechanism to the recurrent neural memory unit, which overcame the problems of gradient dispersion and gradient explosion.

Both LSTM and GRU models can obtain the long-distance context information. LSTM only considers the previous context of the forward time step, but the context information of the backward time step is equally important in sequence labeling. Therefore, this paper utilizes Bi-directional Long Short-term Memory network (Bi-LSTM) to make full use of the context information of words to obtain the feature representation.

The equations of the LSTM are given as follows:

$$\mathbf{i}_t = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i) \quad (1)$$

$$\mathbf{o}_t = \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o) \quad (2)$$

$$\mathbf{f}_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f) \quad (3)$$

$$\mathbf{g}_t = \tanh(W_g \cdot x_t + U_g \cdot h_{t-1} + b_g) \quad (4)$$

$$\mathbf{c}_t = \mathbf{i}_t \otimes \mathbf{g}_t + \mathbf{f}_t \otimes \mathbf{c}_{t-1} \quad (5)$$

$$\mathbf{h}_t = \mathbf{o}_t \otimes \tanh \mathbf{c}_t \quad (6)$$

where \mathbf{i}_t , \mathbf{o}_t , \mathbf{f}_t and \mathbf{c}_t are the components of the LSTM unit, which are input gate, output gate, forget gate and cell unit respectively. And \mathbf{g}_t is the extracted feature vector, \mathbf{h}_t is the output of hidden unit at each time step.

The illustration of the Bi-directional LSTM is shown in Fig. 2.

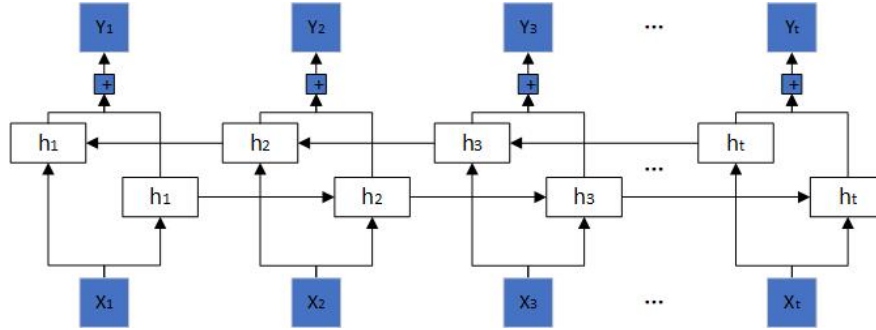


Figure 2: The illustration of the Bi-directional LSTM

The Bi-LSTM neural network is composed of a forward LSTM and a backward LSTM. Forward LSTM encodes sentence sequence, then backward LSTM encodes the sentence sequence in reverse sequence. The hidden states of each time step are expressed as follows:

$$\tilde{\mathbf{h}}_t = LSTM(x_t, \tilde{\mathbf{h}}_{t-1}) \quad (7)$$

$$\tilde{\mathbf{h}}_t = LSTM(x_t, \tilde{\mathbf{h}}_{t-1}) \quad (8)$$

The LSTM hidden states of each time step obtained by forward context and backward context are connected to obtain the representation of the sentence:

$$\mathbf{h}_t = [\vec{\mathbf{h}}_t, \overleftarrow{\mathbf{h}}_t] \quad (9)$$

3.3 Argument recognition layer

Part-of-speech tagging and semantic role labeling share the encoding layer, which makes the model retain a large amount of part-of-speech information. How to find the predicate-argument structure in a sentence is important for semantic role labeling task, which can obtain the most appropriate argument for the predicate.

In order to get the predicate-argument structure in the sentence, we add an argument recognition layer in the training process of main task, which effectively achieves the selection of argument-related information by predicates with the help of the attention mechanism.

Attention mechanism can effectively select the more important information for the current task from the miscellaneous information. In recent years, attention mechanism has been widely used in many fields of deep learning, such as image processing, Chinese question classification [Liu, Yang, Lv et al. (2019)] and sentiment analysis [Zeng, Dai, Li et al. (2019)].

The weight coefficient of attention is calculated as follows:

$$\alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^n \exp(e_j)} \quad (10)$$

The calculation of e_i uses the hidden state h_i of each time step in Bi-LSTM and the representation of the predicate V_p :

$$e_i = v^T \tanh(W_i h_i + W_p V_p + b_i) \quad (11)$$

The final weighted sum of the attention is:

$$c_i = \sum_{i=1}^n \alpha_i h_i \quad (12)$$

The word representation c_i selected by the predicate and the hidden state h_i encoded by Bi-LSTM are concatenated together as the final representation of each word.

3.4 Auxiliary deep neural network model

The auxiliary deep neural model (ANNM) proposed in this paper is based on the joint learning framework, which is realized through the joint learning of semantic role labeling and part-of-speech tagging. Among them, the main structure of the model is multi-layer Bi-LSTM network. And the workflow of the ANNM is shown in Tab. 1.

Table 1: The workflow of the ANNM

Workflow Auxiliary Deep Neural Network Model	
Input: word, predicate	
Output: main loss of the model $Loss_m$	
1.	Initialization map the word in the sentences into word vector V_w , and map the predicate into word vector V_p
2.	Take the word vector V_w as the initial input of the model X_i
3.	Input X_i into three-layer Bi-LSTM
4.	for each i in epoch do
5.	$h_i = Bi-LSTM(X_i)$
6.	if $i \% n \neq 0$ then
7.	$c_i = Attention(h_i, V_p)$
8.	$h'_i = [c_i, h_i]$
9.	$Loss_m = C_1(h'_i, label_{role})$
10.	Back propagation
11.	else
12.	$Loss_a = C_2(h_i, label_{pos})$
13.	Back propagation
14.	end if
15.	end for
16.	return main loss $Loss_m$

In our model, the words in the sentence are first mapped to the real-value vectors V_w , and then the vectors are used as the initial input of the model. The model uses the three-layer Bi-LSTM to obtain context information of the words in the sentence. Dropout is introduced into the multi-layer Bi-LSTM to alleviate the over-fitting in the training process.

The main task of the model is semantic role labeling, and the auxiliary task is part-of-speech tagging. The two tasks share multi-layer Bi-LSTM structures. And an additional argument recognition layer is added to the main task to obtain semantic dependency information between predicates and arguments. In the process of training, the main task and the auxiliary task are trained alternately, but the training times of the main task and the auxiliary task are different. First, the main task is trained many times to get the training loss $Loss_m$, and the parameters of model are adjusted by back propagation algorithm, and then the auxiliary task is trained once to get the training loss $Loss_a$. The

training process is repeated in turn until the loss value of the main task hovered near a fixed value.

In the prediction phase, the auxiliary task directly uses the hidden state of three-layer Bi-LSTM to complete part-of-speech tagging. However, the main task first connects the output of the argument recognition layer with the hidden state of Bi-LSTM as the final representation, and then enters the spliced vector into the linear layer. Finally, the model uses the SoftMax layer to get categories of maximum probability.

3.5 Training

Our method is used to complete the task of sequence labeling by transforming the problem of multi-label into the one of multi-class classification. Then the output of the neural network is an array of C dimensions, and each dimension in the array corresponds to the probability of a category.

The output of the model is transformed into a probability distribution by using the SoftMax. The equation of SoftMax function is shown as follows, and the C is the number of categories in the sequence.

$$\text{soft max}(h_i) = \frac{\exp(h_i)}{\sum_i^C \exp(h_i)} \quad (13)$$

The loss of the model in this paper includes the cross-entropy loss of semantic role labeling and the cross-entropy loss of part-of-speech tagging, and the calculation formulas are show below:

$$Loss_m = -\sum_{t \in N} \sum_{i \in T} \sum_{c=1}^{C_1} p_c(i) \cdot \log(q_c(i)) \quad (14)$$

$$Loss_a = -\sum_{t \in N} \sum_{i \in T} \sum_{c=1}^{C_2} p_c(i) \cdot \log(q_c(i)) \quad (15)$$

where N is the training data, C_1 , C_2 are the number of semantic role categories and part-of-speech categories respectively; t represents the input sentence; T is the length of each sentence; $p_c(i)$ is the probability that each word i is classified as C ; $q_c(i)$ determines whether class C is the correct label. In the training process, the model calculates the derivative of loss function through back propagation algorithm, and the parameters are updated by the stochastic gradient descent method.

4 Experiment

4.1 Experimental dataset and evaluation method

We choose the shared task dataset of CoNLL2005, which is based on the predicate-argument structure annotation of PropBank. Section 2 to 21 of the West Street Journal (WSJ) are used as the training set, Section 24 of the WSJ as the validation set, Section 23 of the WSJ and three sections of the Brown corpus as the test set.

In order to compare with the results of previous methods, we adopt the F1-score as the evaluation standard in our experiments.

4.2 Parameter settings

In the initialization of our model, all weight matrices are randomly initialized to orthogonal matrices, and all the bias are initialized to 0. The number of Bi-LSTM hidden sizes is 300, the minibatch size is 128, the learning rate is 0.0001, and the decay factor of learning rate is 0.98. The optimization of model uses stochastic gradient descent (SGD) and Adam to adjust automatically. The cross-entropy loss with L2 regularization is selected as the loss function, and decay factor of weight coefficient is 0.0002. The final hyper-parameters are shown in Tab. 2.

Table 2: The hyperparameter settings of the model

Parameter	Value
Word embedding size	300
Bi-LSTM hidden size	300
Bi-LSTM layer	3
Learning rate	0.0001
Dropout	0.5
Minibatch size	128

4.3 Comparative analysis with/without auxiliary task

In order to evaluate the effect of auxiliary task and argument recognition layer in the proposed model. First, we define the SRL model using only multi-layer Bi-LSTM as the LSTM-SRL; Second, we add auxiliary task to the LSTM-SRL for joint learning (LSTM-SRL & Aux-Task); Then, we introduce the argument recognition layer into the LSTM-SRL for exploring the validity of predicate-argument structure (LSTM-SRL & ARL); Finally, we add auxiliary task and argument recognition layer to the LSTM-SRL model, which is the ANNM proposed in this paper. The specific results are shown in Tab. 3.

Table 3: The results with/without Auxiliary task on the CoNLL2005

Model	Dev			Test (WSJ)			Test (Brown)		
	P	R	F1	P	R	F1	P	R	F1
LSTM-SRL	0.849	0.832	0.840	0.859	0.849	0.854	0.783	0.753	0.768
LSTM-SRL & Aux-Task	0.847	0.851	0.849	0.878	0.869	0.873	0.792	0.772	0.782
LSTM-SRL & ARL	0.851	0.857	0.854	0.887	0.875	0.881	0.792	0.778	0.785
ANNM	0.859	0.869	0.864	0.891	0.889	0.890	0.799	0.789	0.794

When adding the auxiliary task to the LSTM-SRL, the results show that the F1 values of the model are improved by 1.9% and 1.4% respectively in WSJ and Brown test sets. At

the same time, the results indicate that ANNM is effective in the joint training of part-of-speech tagging and semantic role labeling. Under the same condition of using three Bi-LSTM layers, the alternating training of the two tasks enables the model to obtain the semantic relationship between the tasks, thus sharing the effective features by the encoding layer.

When adding the argument recognition layer (ARL) to the LSTM-SRL, the F1 values are improved by 2.7% and 1.7% respectively in WSJ and Brown test sets. Among them, the information related to the arguments is selected by predicates, and the context information is obtained by the multi-layer Bi-LSTM. When two kind of information are concatenated, the results show that they can effectively improve the accuracy of main task. At the same time, the results verify the importance of predicates in the task and the rationality of adding argument recognition layer in the main task. Compared with the results of adding the auxiliary task, the results of adding ARL are better, which also proves that the predicate is more important information.

When adding both the auxiliary task (Aux-Task) and the argument recognition layer (ARL) to the LSTM-SRL, the F1 values of the model are improved by 3.6% and 2.6% respectively in WSJ and Brown test sets. The results show that the addition of auxiliary task (Aux-Task) and argument recognition layer (ARL) in the proposed ANNM is effective. Moreover, the ANNM can effectively utilize the information in the shared encoding layer and structural information of predicate-argument in the argument recognition layer.

4.4 Comparative analysis with other models

To demonstrate the advantages of the proposed model, we select five other models to compare with our method in Tab. 4.

Table 4: The results of different models on the CoNLL2005

Model	Dev			Test (WSJ)			Test (Brown)		
	P	R	F1	P	R	F1	P	R	F1
He et al. (2017) PoE	0.831	0.824	0.827	0.850	0.843	0.846	0.749	0.724	0.736
He et al. (2018)	-	-	0.816	0.812	0.839	0.825	0.697	0.719	0.708
He et al. (2018)+ELMo	-	-	0.853	0.848	0.872	0.860	0.739	0.784	0.761
Angel Daza et al. (2018)	-	-	0.773	-	-	0.792	-	-	0.684
Wang et al. (2019)	-	-	-	-	-	0.882	-	-	0.793
ANNM	0.859	0.869	0.864	0.891	0.889	0.890	0.799	0.789	0.794

He et al. [He, Lee, Lewis et al. (2017)] used 8-layer Bi-LSTM structure with constrained decoding. The model could obtain long-distance information of predicate-argument, but it needed constrained decoding to overcome the structural inconsistencies. Our ANNM only uses the three-layer stacked Bi-LSTM layer to learn the feature representation. While avoiding the over-fitting problem of the multi-layer neural network, the two tasks can still achieve better results through the sharing feature. He et al. [He, Lee, Levy et al.

(2018)] proposed an end-to-end method for joint prediction of predicates and argument spans, which overcame some key limitation of semi-Markov and BIO tagging models, but it did not do well in ensuring global consistency. Our ANNM method directly uses the hidden state of each time step to decode one-to-one, making use of the full text information, while preserving the global consistency. Daza et al. [Daza and Frank (2018)] converted the sequence labeling task into the task of sequence to sequence model, but it still faced the problem that the generated sequence couldn't match the source sequence exactly, and the accuracy of the generated sequence was low. Wang et al. [Wang, Johnson, Wan et al. (2019)] introduced the external syntactic information as part of the input. However, the results of syntactic parsing with low accuracy also caused the problem of error propagation. Without the support of additional syntactic information, ANNM based on multi-task learning can still achieve excellent results. Finally, the results show that our model has better performance than the existing models. And our model obtains the F1 value of 89.0% on the WSJ test set of CoNLL2005, which is 0.8% higher than the existing best model.

5 Conclusion

In this paper, we propose an auxiliary deep neural network model (ANNM) for joint learning of part-of-speech tagging and semantic role labeling. The model takes semantic role labeling as the main task and part-of-speech tagging as the auxiliary task. And the main task and auxiliary task are trained alternately, which can improve the performance of semantic role labeling. Based on the framework of multi-task learning, the model uses the multi-layer stacked Bi-LSTM to accomplish two different tasks. At the same time, an argument recognition layer is added to the training process of the main task, which enables the predicate to select the information related to the arguments. Finally, the excellent F1 value is obtained on the WSJ test set of CoNLL2005, which exceeds the best results of all existing models.

This model has researched the effect of part-of-speech and predicate-argument structural information on semantic role labeling. However, it is worth exploring to the influence of syntactic information and selective preference information in semantic role labeling.

Acknowledgement: We would like to thank all editors for their detailed review and useful feedback, and we would also like to thank all reviewers for their constructive comments and suggestions for revision.

Funding Statement: The work of this article is supported by Key Scientific Research Projects of Colleges and Universities in Henan Province (Grant No. 20A520007), National Natural Science Foundation of China (Grant No. 61402149).

Conflicts of Interest: No conflict of interest exists in the submission of this manuscript, and manuscript is approved by all authors for publication.

References

- Bastings, J.; Titov, I.; Aziz, W.; Marcheggiani, D.; Sima'an, K.** (2017): Graph convolutional encoders for syntax-aware neural machine translation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1957-1967.
- Cai, J.; He, S.; Li, Z.; Zhao, H.** (2018): A full end-to-end semantic role labeler, syntax-agnostic over syntax-aware? *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2753-2765.
- Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F. et al.** (2014): Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1724-1734.
- Collobert, R.; Weston, J.** (2008): A unified architecture for natural language processing: deep neural networks with multitask learning. *Proceedings of the 25th International Conference on Machine learning*, pp. 160-167.
- Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K. et al.** (2011): Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, vol. 12, pp. 2493-2537.
- Daza, A.; Frank, A.** (2018): A sequence-to-sequence model for semantic role labeling. *Proceedings of the 3rd Workshop on Representation Learning for NLP*, pp. 207-216.
- FitzGerald, N.; Täckström, O.; Ganchev, K.; Das, D.** (2015): Semantic role labeling with neural network factors. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 960-970.
- Gildea, D.; Jurafsky, D.** (2002): Automatic labeling of semantic roles. *Computational Linguistics*, vol. 28, no. 3, pp. 245-288.
- He, L.; Lee, K.; Levy, O.; Zettlemoyer, L.** (2018): Jointly predicting predicates and arguments in neural semantic role labeling. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 364-369.
- He, L.; Lee, K.; Lewis, M.; Zettlemoyer, L.** (2017): Deep semantic role labeling: what works and what's next. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 473-483.
- Idrees, M.; FitzGerald, J. M.** (2015): Vocal cord dysfunction in bronchial asthma a review article. *Journal of Asthma*, vol. 52, no. 4, pp. 327-335.
- Liu, J.; Yang, Y. H.; Lv, S. Q.; Wang, J.; Chen, H.** (2019): Attention-based BiGRU-CNN for Chinese question classification. *Journal of Ambient Intelligence and Humanized Computing*, <https://doi.org/10.1007/s12652-019-01344-9>.
- Marcheggiani, D.; Titov, I.** (2017): Encoding sentences with graph convolutional networks for semantic role labeling. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1506-1515.
- Mikolov, T.; Chen, K.; Corrado, G.; Dean, J.** (2013): Efficient estimation of word representations in vector space. arXiv:1301.3781.
- Ouchi, H.; Shindo, H.; Matsumoto, Y.** (2018): A span selection model for semantic role labeling. *Proceedings of the Conference on Empirical Methods in Natural Language*

Processing, pp. 1630-1642.

Pal, R.; Sharma, D. M. (2019): A dataset for semantic role labelling of Hindi-English code-mixed tweets. *Proceedings of the 13th Linguistic Annotation Workshop*, pp. 178-188.

Palmer, M.; Babko-Malaya, O.; Bies, A.; Diab, M. T. (2008): A pilot arabic propbank. in LREC. *Proceedings of the International Conference on Language Resources and Evaluation*.

Punyakanok, V.; Roth, D.; Yih, W. T. (2008): The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, vol. 34, no. 2, pp. 257-287.

Roth, M.; Lapata, M. (2016): Neural semantic role labeling with dependency path embeddings. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1192-1202.

Shi, C.; Liu, S.; Ren, S.; Feng, S.; Li, M. et al. (2016): Knowledge-based semantic embedding for machine translation. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 2245-2254.

Swayamdipta, S.; Thomson, S.; Lee, K.; Zettlemoyer, L.; Dyer, C. et al. (2018): Syntactic scaffolds for semantic structures. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 3772-3782.

Wang, Y.; Johnson, M.; Wan, S.; Sun, Y.; Wang, W. (2019): How to best use syntax in semantic role labelling. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5338-5343.

Yin, L. B.; Meng, X.; Li, J. X.; Sun, J. G. (2019): Relation extraction for massive news texts. *Computers, Materials & Continua*, vol. 60, no. 1, pp. 275-285.

Zeng, D. J.; Dai, Y.; Li, F.; Wang, J.; Sangaiah, A. K. (2019): Aspect based sentiment analysis by a linguistically regularized CNN with gated mechanism. *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 5, pp. 3971-3980.

Zhang, Z.; Zhao, H.; Qin, L. (2016): Probabilistic graph-based dependency parsing with convolutional neural network. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1382-1392.

Zhou, J.; Xu, W. (2015): End-to-end learning of semantic role labeling using recurrent neural networks. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 1, pp. 1127-1137.