A Sequence-to-Action Architecture for Character-based Chinese Dependency Parsing with Status History

Hang Liu[1,2], Yujie Zhang[2], Meng Chen[1], Jinan Xu[2], and Yufeng Chen[2]

[1] JD AI, Beijing, China
[2] School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China
liuhang55@jd.com, yjzhang@bjtu.edu.cn, chenmeng20@jd.com, jaxu@bjtu.edu.cn, chenyf@bjtu.edu.cn

Abstract. Character-based Chinese dependency parsing jointly learns Chinese word segmentation, POS tagging and dependency parsing to avoid the error propagation problem of pipeline models. Recent works on this task only rely on a local status for prediction at each step, which is insufficient for guiding global better decisions. In this paper, we first present a sequence-to-action model for character-based dependency parsing. In order to exploit decision history for prediction, our model tracks the status of parser particularly including decision history in the decoding procedure by employing a sequential LSTM. Additionally, for resolving the problem of high ambiguities in Chinese characters, we add position-based character embeddings to exploit character information with specific contexts accurately. We conduct experiments on Penn Chinese Treebank 5.1 (CTB-5) dataset, and the results show that our proposed model outperforms existing neural network system in dependency parsing, and performs preferable accuracy in Chinese word segmentation and POS tagging.

Keywords: Character-based Chinese dependency parsing · Decision history · Character information.

1 Introduction

Character-based Chinese dependency parsing is a joint model for Chinese word segmentation, POS tagging and dependency parsing, which aims to prevent the error propagation problem of pipeline models [1]. The model is usually implemented using transition-based framework and is viewed as a transition sequence decision task [2]. Existing approaches for joint model are categorized into two types: conventional discrete feature-based approaches[1, 3, 4] and neural network-based models[5, 6]. Feature-based models tackle the effort in hand-crafting effective feature combination and define large feature templates to capture features, and maintain state-of-the-art performance. Neural network-based models uses dense vectors and LSTM to reduce the cost of feature engineering, and achieves competitive performance in all three tasks when using very few features. Despite of existing approaches success, there are still two problems.

The first problem is insufficient history information. In transition-based framework, these transition actions modify the current parser state after each decision, and the previous parser state is discarded. Recent evidence on word-level dependency parsing [7] and natural language inference [8] reveals that history information of actions taken and parser state is of crucial important. However, feature-based joint models and neural network-based joint models rely on local parser state for the action prediction at that point in transition sequence. This prevents model from exploring sufficient context information of parser state for making better decisions.

The second problem is the ambiguity of Chinese character. A Chinese simplified character may be derived from two traditional characters and therefore becomes more ambiguous. For example, the character "发" in the "头发" (hair) and " 发财" (make a fortune) has different meanings that are the " 髮" (hair) and "發" (make) respectively. It is worth noting that when representing the meaning of "髮" (hair), the character "发" is located at the end of the word as an object, while when representing the meaning of "發" (make), the character "发" is located at the beginning of the word as a verb. Most previous models rely on single one-hot representation [1, 3, 4] or distributed representation [5, 6] of character. However, existing character embedding did not catch such position information for disambiguation.

In order to address these two problems, we take advantage of the seq2seq [9] to capture history information and propose a sequence-to-action joint model. In the encoder of the model, we use attention mechanism and position-based character embeddings to capture character information with specific contexts. In the decoder, a sequential LSTM is used to track history information, including previous transition actions taken by the parser and all other previous parser state in the decoding procedure. We evaluate our model on CTB-5 dataset, and find that it significantly outperforms the bi-LSTM [5] joint model in each task and compares favorably with the state-of-the-art feature engineering joint models.

Our contributions are summarized as follows:

- We first conduct encoder-decoder architecture for Character-based Chinese dependency parsing, which allows the model to track the history of decision taken by parser.
- The sequence-to-action joint model incorporates position-based character embedding to capture more exact meanings of a character within specific contexts.
- We evaluate our model on CTB-5 dataset and the results show that our model outperforms existing bi-LSTM joint model.

2 Sequence-to-Action Joint Model

This section describes our proposed model. Transition-based parsing is a task of predicting a series of transition actions $y \in Y$ for a given sentence, where

Y = {SH, AP, RR, RL}. Following the arc-standard algorithm [10], our model consists of one buffer and one stack. The buffer contains character in the input sentence, and the stack contains partially-built dependency subtrees.

The subtrees in the stack are formed by the following transition actions:

– SH(pos): Shift the first character of the buffer to the top of the stack as a new word, and the POS tag (pos) is attached to the word.
– AP: Append the first character of the buffer to the end of the top word of the stack.
– RR: Reduce the right word of the top two words of the stack, and make the right child node of the left word.
– RL: Reduce the left word of the top two words of the stack, and make the left child node of the right word.

Given one sentence $S = (c_1, ..., c_i, ..., c_n)$, the aim of the model is to predict the groundtruth of actions:

$$y^* = \arg\max \mathit{\Pi}_{t=1}^{|y|} P(y_t | parser_t) \tag{1}$$

where $P(.)$ is the sequence-to-action joint model here. y_t and $parser_t$ are the action and parser state at time step t respectively. $|y|$ is the action number.

The sequence-to-action architecture we proposed for Chinese parsing is illustrated as in Figure 1, which consists of two components: **an encoder layer** converts the input sentence into distributed representation; **a decoder layer** captures parser state and tracks history information to make decisions.

2.1 Encoder Layer

Given input sentence $S = (c_1, ..., c_i, ..., c_n)$, the encoder layer first converts them into vectors $(v_1, ..., v_i, ..., v_n)$ by looking up M and the corresponding position-based character vectors $(v_i^s, v_i^b, v_i^m, v_i^e)$ are retrieved by looking up PM, where M and PM are the embedding tables. b, m, e and s are the position of character within a word, representing *begin*, *middle*, *end* and *single* respectively.

In the work of Chen et al. [11], one of position-based character vectors is selected according to the position of character in word. However, in our work the position of character is unknown before parsing. Different from their work, we calculate the character vector from all of the position-based character vectors by using the attention mechanism and taking a context of K length window as attention. Formally, the contextual character vectors could be computed as:

$$w_i = \frac{1}{2K+1} \sum_{k=i-K}^{k=i+K} v_k \tag{2}$$

$$u_i^p = w_i^T W v_i^p + \langle U^T, w_i \rangle + \langle V^T, v_i^p \rangle \tag{3}$$

$$a_i^p = softmax(u_i^p) \tag{4}$$

$$v'_i = \sum_{p \in \{b,m,e,s\}} a_i^p \cdot v_i^p \tag{5}$$

where w_i is the context embedding, considering a local window of character embeddings. We adopt the biaffine attention mechanism for attention score function [12]. Here, W, U, V are trainable parameters, and p is the position of character.

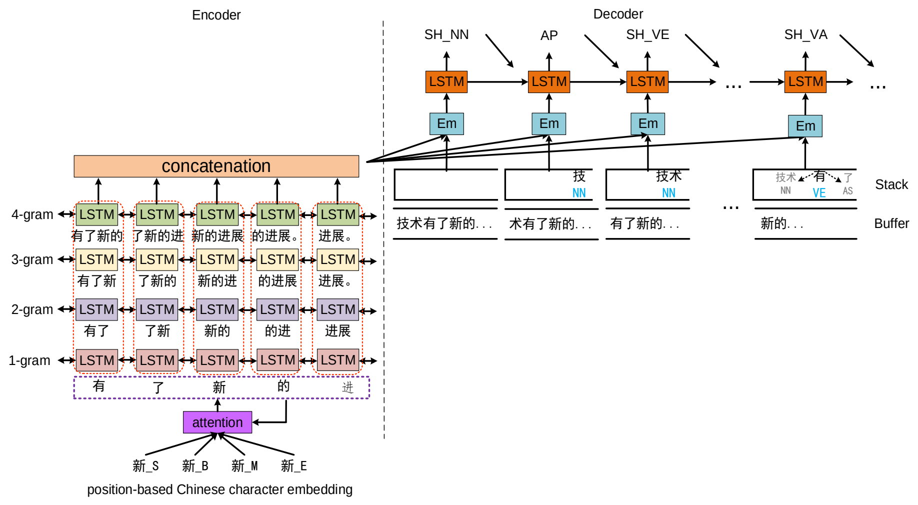

Fig. 1. Our sequence-to-Action Architecture for character-based Chinese dependency parsing. The dashed box is the feature extractor, capturing parser state. The encoder uses position-based embedding and bidirectional LSTM to produce the semantic representation. After that, the decoder captures parser state and tracks history information to predict transition actions.

Then four bidirectional LSTMs are used to capture uni-gram, bi-gram, tri-gram and four-gram character strings in the input sentence respectively. All n-gram inputs to bidirectional LSTM are given by looking up M or character string embeddings [5] by contextual character vectors. The hidden outputs of four bidirectional LSTMs are concatenated to produce the semantic representation for each character in S.

2.2 Decoder Layer

In order to capture all of the state information and all previous decisions taken by the parser, we employ a sequential LSTM to maintain a history of the portion of the sentence that has been processed so far. In this layer, a feature function is used to extract the feature representations of characters from the encoder and built subtrees, including n-gram features, POS features and dependency structure features (encoded by using Tree-LSTM [13]) of the top three items of the stack, and n-gram feature of first item of the buffer. The feature representations extracted by feature function is a local state. The sequential LSTM takes inputs from the local state and previous decision at each time step. It can be formulated as:

$$h_t = LSTM(parser_t, y_{t-1}, h_{t-1}) \tag{6}$$

$$P(y_t|parser_t) = softmax(W_1 h_t + b_1) \tag{7}$$

where W_1, b_1 are trainable parameters, y_{t-1} is the transition action at the last time step.

2.3 Training

We train the sequence-to-action joint model with the objective function for greedy training, which can be formulated as:

$$J(\theta) = -\frac{1}{N}\sum_{t=1}^{t=N} \log P(y_t|parser_t) + \frac{\lambda}{2} \parallel \theta \parallel^2 \tag{8}$$

where N is the number of actions in one sentence, θ denotes all the trainable parameters of our model. Adam [14] is adopted as optimizer.

3 Experiments

3.1 Experiment Settings

In this section, we evaluate our parsing model on the Penn Chinese Treebank 5.1 (CTB-5), following the splitting of Jiang et al. [15]. The dataset statistic is shown in Table 1. All word and character embeddings are initialized with 200-dimension word2vec vectors [16]. The POS and action embeddings are initialized to random values with 200 dimensions. The bidirectional LSTMs' hidden states is 200 dimensions and LSTM's hidden state is 400 dimensions. We set the window size K=2 and the initial learning rate is 0.001. We also employ a dropout strategy [17] to avoid over-fitting and the dropout rate is set to 0.33. The batch size is set to 32.

Table 1. Statistics of dataset.

	sentence	**word**	**oov**
Training	18k	494k	*
Development	350	6.8k	553
Test	348	8.0k	278

3.2 Results

We use word-level F1 score to evaluate word segmentation, POS tagging and dependency parsing, following previous works [1, 3–6]. Dependency parsing task is evaluated with the unlabeled attachment scores excluding punctuation. The output of POS tags and dependencies cannot be correct unless the corresponding words are segmented correctly.

Table 2 lists the comparison results of our joint model compared with other state-of-the-art joint models. We can see that out proposed model can significantly outperform the basic bi-LSTM joint model with 0.1%(word segmentation), 0.7%(POS) and 1.44%(dependency) respectively for improvements. This demonstrates the effectiveness of our proposed sequence-to-action architecture for neural character-based Chinese dependency parsing. Besides, our model achieves better F1 score than the neural joint models of Kurita17 and Li18 on dependency parsing. Although the performance is slightly lower than those of feature engineering-based models, like Hatori12, Zhen14 and Zhang14, our model achieves competitive performance. The performance improvement of our model is due to make full use of context features and parser state by employing encoder and decoder. It also suggests that our model can combines with feature engineering for better performance.

Table 2. Comparison with previous models on the CTB dataset.

Models	Method	Representation	Seg	POS	Dep
Hatori12 [1]	beam	large feature set(66, sparse)	97.75	94.33	81.56
Zhang14 [3]	beam	large feature set(101, sparse)	97.67	94.28	**81.63**
Zhen14 [4]	beam	large feature set(53, sparse)	97.52	93.93	79.55
Kurita17 [5]	greedy	large feature set(50, dense)	**98.24**	**94.49**	80.15
Li18 [6]	greedy	3LSTM vectors	96.64	92.88	79.44
bi-LSTM [5]	greedy	4LSTM vectors	97.72	93.12	79.03
Our	greedy	(4LSTM, 3POS, 3subtree) vectors	**97.88**	**93.82**	**80.47**

3.3 Effect of Components

Table 3. Effects of the different components.

Models	Seg	POS	Dep
bi-LSTM joint model	97.72	93.12	79.03
+decoder LSTM	97.51	93.53	79.62
+position embedding	97.82	93.42	80.02
+POS,subtree	**97.88**	**93.82**	**80.47**

We perform some ablation experiments to analyze the effect of the different components on our models. As illustrated in Table 3, the first row is the baseline bi-LSTM joint model [5]. Compared to the baseline model, we use sequential LSTM as decoder instead of MLP for prediction, and F1 score increases by 0.41% and 0.59% on POS and dependency parsing respectively. It proves the decoder with LSTM can capture the helpful information from previous state and decisions, which is more effective for POS and dependency parsing. By adding the position-based character embedding, the word segmentation and dependency parsing performances are increased to 97.82% and 80.02% respectively. It is because the character representation is more accurate by capture meanings of character within specific local contexts. Elmo [18] and Bert [19] have been well known as pre-trained language model for acquiring contextual character vectors. However, our current condition of computing power could not support such complicated training task. We will conduct the comparison with them in the future. In this paper, we evaluated the contribution of position embedding to our model. Besides, our model achieves further improvement in each task by additionally adopting POS features and subtree features.

4 Related Work

Transition-based joint model for Chinese word segmentation, POS tagging and dependency parsing was proposed by Hatori et al. [1]. Zhang et al. [3] and Zhen et al. [4] extended this work by adding word-structure features to extract intra-word dependencies. Kurita et al. [5] proposed the first embedding-based joint parsing model and used the character string embeddings to replace incomplete or unknown words embeddings. All above mentioned works relied heavily on handcrafted features and it was a hard and time consuming task to define a good feature-set. In contrast to these, the neural parsing model we presented in this work only used a little feature and achieves comparable performance. Besides, Kurita et al. [5] also explored bi-LSTM models to avoid the detailed feature engineering, but they only extracted local state to make decisions and neglected parser history information. Li et al. [6] provided rich character-level POS and dependency annotations to better understanding deeper structure of Chinese words.

5 Conclusion

In this paper, we propose a novel sequence-to-action joint model for character-based dependency parsing to track history information of parser in the decoding procedure. Besides, we use position-based character embeddings to capture exact character meanings within specific contexts. Experimental results demonstrate that our proposed model significantly outperforms the existing neural models for joint paring, and achieves comparable performance with the state-of-the-art joint models.

In the future, we will expand the scale of the experiment and further verify the effectiveness of the proposed method. In addition, we further explore better way to learning character representations, such as Elmo and Bert.

Acknowledgments. The authors are supported by the National Nature Science Foundation of China (61876198, 61370130 and 61473294).

References

1. Hatori, J., Matsuzaki, T., Miyao, Y., Tsujii, J. I. Incremental joint approach to word segmentation, POS tagging, and dependency parsing in Chinese. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, pp. 1045-1053. Association for Computational Linguistics (2012)
2. Nivre, J. An efficient algorithm for projective dependency parsing. In Proceedings of the Eighth International Conference on Parsing Technologies, pp. 149-160 (2003).
3. Zhang, M., Zhang, Y., Che, W., Liu, T. Character-level Chinese dependency parsing. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers, Vol. 1, pp. 1326-1336 (2014).
4. Guo, Z., Zhang, Y., Su, C., Xu, J., Isahara, H. Character-level dependency model for joint word segmentation, POS tagging, and dependency parsing in Chinese. IEICE TRANSACTIONS on Information and Systems, 99(1), 257-264 (2016).
5. Kurita, S., Kawahara, D., Kurohashi, S. Neural joint model for transition-based Chinese syntactic analysis. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers, pp. 1204-1214 (2017).
6. Li, H., Zhang, Z., Ju, Y., Zhao, H. Neural character-level dependency parsing for Chinese. In Thirty-Second AAAI Conference on Artificial Intelligence (2018).
7. Dyer, C., Ballesteros, M., Ling, W., Matthews, A., Smith, N. A. Transition-Based Dependency Parsing with Stack Long Short-Term Memory. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Volume 1: Long Papers, Vol. 1, pp. 334-343 (2015).
8. Bowman, S. R., Gauthier, J., Rastogi, A., Gupta, R., Manning, C. D., Potts, C. A Fast Unified Model for Parsing and Sentence Understanding. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers, Vol. 1, pp. 1466-1477 (2016).
9. Sutskever, I., Vinyals, O., Le, Q. V. Sequence to sequence learning with neural networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2, pp. 3104-3112. MIT Press (2014).
10. Nivre, J. Incrementality in deterministic dependency parsing. In Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together, pp. 50-57. Association for Computational Linguistics (2004).
11. Chen, X., Xu, L., Liu, Z., Sun, M., Luan, H. Joint learning of character and word embeddings. In Proceedings of the 24th International Conference on Artificial Intelligence, pp. 1236-1242. AAAI Press (2015).
12. Dozat, T., Manning, C. D. Deep biaffine attention for neural dependency parsing. arXiv preprint arXiv:1611.01734 (2016).
13. Tai, K. S., Socher, R., Manning, C. D. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In Proceedings of the 53rd

Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Volume 1: Long Papers, Vol. 1, pp. 1556-1566 (2015).

14. Kingma, D. P., Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
15. Jiang, W., Huang, L., Liu, Q., Lv, Y. A cascaded linear model for joint chinese word segmentation and part-of-speech tagging. In In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (2008).
16. Mikolov, T., Chen, K., Corrado, G., Dean, J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013).
17. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958 (2014).
18. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1: Long Papers, pp. 2227-2237 (2018).
19. Devlin, J., Chang, M. W., Lee, K., Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).