Segmenting Natural Language Sentences via Lexical Unit Analysis

Yangming Li, Lemao Liu, Shuming Shi
Tencent AI Lab
Shenzhen, China
{newmanli,redmondliu,shumingshi}@tencent.com

Abstract

The span-based model enjoys great popularity in recent works of sequence segmentation. However, each of these methods suffers from its own defects, such as invalid predictions. In this work, we introduce a unified span-based model, lexical unit analysis (LUA), that addresses all these matters. Segmenting a lexical unit sequence involves two steps. Firstly, we embed every span by using the representations from a pretraining language model. Secondly, we define a score for every segmentation candidate and apply dynamic programming (DP) to extract the candidate with the maximum score. We have conducted extensive experiments on 3 tasks, (e.g., syntactic chunking), across 7 datasets. LUA has established new state-of-the-art performances on 6 of them. We have achieved even better results through incorporating label correlations.[1]

1 Introduction

Plenty of tasks in natural language understanding (NLU), such as syntactic chunking, are essentially a sequence segmentation problem, which partitions a sequence of lexical units into multiple labeled segments. A classical approach to sequence segmentation is to cast it into a sequence labeling task using IOB tagging scheme (Ma and Hovy, 2016; Liu et al., 2019c; Luo et al., 2020). Every token in a sentence, according to its position in the corresponding segment, is labeled with a tag (e.g., B-PER). A representative work is Bidirectional LSTM-CRF (Huang et al., 2015).

Recently, there is a surge of interest in developing span-based models (Cai and Zhao, 2016; Zhai et al., 2017; Li et al., 2020a; Yu et al., 2020; Li et al., 2021). They regard spans rather than tokens as the basic units for labeling. For example, Li et al. (2020a) model named entity recognition (NER) as machine reading comprehension (MRC) (Seo et al., 2017), where entities are extracted as retrieving answer spans. While span-based models have achieved promising performances, they are locally normalized at span level, and therefore suffered from the label bias problem (Lafferty et al., 2001). Moreover, some of them (Yu et al., 2020; Li et al., 2021) rely on heuristic rules to correct invalid predictions (e.g., span conflicts between two entities). Early span-based models (Andrew, 2006; Kong et al., 2016; Ye and Ling, 2018; Liu et al., 2019a) based on Semi-Markov CRF (Sarawagi and Cohen, 2005) adopts dynamic programming (DP) (Bellman, 1966) to search for the optimal segmentation of a sentence. Unlike their counterparts (Clark et al., 2018; Akbik et al., 2018; Devlin et al., 2019; Li et al., 2021), these methods all train the sentence encoders from scratch, without exploiting the knowledge from unlabeled corpora. Hence, none of them is even competitive with current best sequence labeling model.

In this paper, we propose lexical unit analysis (LUA), a unified and effective span-based model that circumvents all above problems. Our segmentation of a natural language sentence contains two steps. Firstly, we utilize BERT (Devlin et al., 2019), a powerful pretraining language model, to get contextualized token representations, and with them we embed every span of the sentence, inspired by the finding that pretraining language models are very robust to rare tokens and the low-resource setting (Liu et al., 2019b). Then, we assign a score to every segmentation candidate and use DP to globally search for the candidate with the maximum score. The score of a segmentation is computed from the segment scores predicted by LUA. We minimize the hinge loss, instead of cross-entropy, to train our models.

We have performed extensive experiments on syntactic chunking, Chinese part-of-speech (POS) tagging, and NER across 7 datasets. Our model

[1] The source code for our work is publicly available at https://github.com/LeePleased/LUA.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 181–187
November 7–11, 2021. ©2021 Association for Computational Linguistics

has achieved state-of-the-art results on 6 of them and performed competitively on the remaining one. Besides, we have obtained slightly better performances by explicitly modeling the label dependencies. We also show that LUA is very efficient in terms of running time.

2 Architecture

We denote an input sequence of lexical units as $\mathbf{x} = [x_1, x_2, \cdots, x_n]$. Output segments are represented as the segmentation $\mathbf{y} = [y_1, y_2, \cdots, y_m]$ with each segment y_k being a triple (i_k, j_k, t_k). n and m are respectively the numbers of lexical units and segments. (i_k, j_k) is a span that corresponds to the phrase $\mathbf{x}_{i_k,j_k} = [x_{i_k}, x_{i_k+1}, \cdots, x_{j_k}]$. t_k is a label from the label space $\mathcal{L}$. A segmentation is valid if all its segments are non-overlapping and fully cover the input sentence.

An example from CoNLL-2003 dataset (Sang and De Meulder, 2003):

$$\begin{aligned} \mathbf{x} = [\text{[SOS]}, \text{NEW}, \text{DELHI}, 1996-08-29] \\ \mathbf{y} = [(1,1,\text{O}), (2,3,\text{LOC}), (4,4,\text{O})] \end{aligned}.$$

[SOS] marks the beginning of a sentence and is inserted in the pre-processing stage.

2.1 Constructing Span Representations

Following advanced models (Luo et al., 2020; Yu et al., 2020; Li et al., 2021), we adopt BERT as the sentence encoder to get the contextualized representation for every token x_i:

$$[\mathbf{h}^w_1, \mathbf{h}^w_2 \cdots, \mathbf{h}^w_n] = \text{BERT}(\mathbf{x}). \quad (1)$$

The representation for a span (i, j) is composed of the representations of its end points:

$$\mathbf{h}^p_{i,j} = \mathbf{h}^w_i \oplus \mathbf{h}^w_j, \quad (2)$$

where $\oplus$ is column-wise vector concatenation.

2.2 Scoring and Solving

Assume $\mathcal{Y}$ is the universal set that contains all the valid segmentation candidates for the input sequence $\mathbf{x}$. Given one of its members $\mathbf{y} \in \mathcal{Y}$, we compute the score $f(\mathbf{y})$ as

$$f(\mathbf{y}) = \sum_{(i,j,t)\in\mathbf{y}} \left(s^c_{i,j} + s^l_{i,j,t}\right), \quad (3)$$

where $s^c_{i,j}$ is the composition score to estimate the feasibility of merging several lexical units $\mathbf{x}_{i,j} = [x_i, x_{i+1}, \cdots, x_j]$ into a segment and $s^l_{i,j,t}$ is the label score to measure how likely the label of this segment is t. Both scores, $s^c_{i,j}$ and $s^l_{i,j,t}$, for a span (i, j) are predicted as

$$\begin{cases} s^c_{i,j} = \left(\mathbf{v}^c\right)^T \tanh(\mathbf{W}^c\mathbf{h}^p_{i,j}) \\ s^l_{i,j,t} = \left(\mathbf{v}^l_t\right)^T \tanh(\mathbf{W}^l\mathbf{h}^p_{i,j}) \end{cases}, \quad (4)$$

where $\mathbf{v}^c$, $\mathbf{W}^c$, $\mathbf{v}^l_t, t \in \mathcal{L}$, and $\mathbf{W}^l$ are learnable parameters.

The prediction of the segmentation candidate of the maximum score can be formulated as

$$\widehat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} f(\mathbf{y}). \quad (5)$$

Since the size of search space $|\mathcal{Y}|$ increases exponentially with the sequence length n, brute-force search to solve this is computationally infeasible. LUA utilizes DP to solve this issue.

DP is a well-known optimization method that addresses a complicated problem by breaking it down into multiple simpler sub-problems in a recursive manner. The relation between the value of the larger problem and the values of its sub-problems is called the Bellman equation.

Sub-problem. In the context of LUA, the sub-problem of segmenting an input unit sequence $\mathbf{x}$ is segmenting one of its prefixes $\mathbf{x}_{1,i}, 1 \le i \le n$. We define g_i as the maximum segmentation score of the prefix $\mathbf{x}_{1,i}$. Under this scheme, we have $\max_{\mathbf{y}\in\mathcal{Y}} f(\mathbf{y}) = g_n$.

The Bellman Equation. The relationship between segmenting a sequence $\mathbf{x}_{1,i}, i > 1$ and segmenting its prefixes $x_{1,i-j}, j \le i-1$ is bridged by the last segments $(i-j+1, i, t)$:

$$\begin{aligned} g_i = \max_{1\le j\le i-1} \Big(g_{i-j} + \\ (s^c_{i-j+1,i} + \max_{t\in\mathcal{L}} s^l_{i-j+1,i,t})\Big) \end{aligned}. \quad (6)$$

To improve the computational efficiency, the last term can be computed beforehand as

$$s^T_{i,j} = \max_{t\in\mathcal{L}} s^l_{i,j,t}, 1 \le i \le j \le n. \quad (7)$$

Hence, the final Bellman equation is

$$g_i = \max_{1\le j\le i-1} \left(g_{i-j} + (s^c_{i-j+1,i} + s^T_{i-j+1,i})\right). \quad (8)$$

The base case is the first token $\mathbf{x}_{1,1} = \text{[[SOS]]}$. We get its score g_1 as $s^c_{1,1} + s^T_{1,1}$.

Method	CTB5	CTB6	CTB9	UD1
Bi-RNN + CRF (Single) (Shao et al., 2017)	94.07	90.81	91.89	89.41
Bi-RNN + CRF (Ensemble) (Shao et al., 2017)	94.38	-	92.34	89.75
Lattice-LSTM (Zhang and Yang, 2018)	95.14	91.43	92.13	90.09
BERT Tagging (Devlin et al., 2019)	96.06	94.77	92.29	94.79
McASP (Tian et al., 2020)	96.60	94.74	94.78	95.50
LUA	**96.81**	**95.36**	**94.93**	**96.02**

Table 1: Experiment results on the four datasets of Chinese POS tagging.

Method	Chunking	NER	
	CoNLL-2000	CoNLL-2003	OntoNotes 5.0
Bi-LSTM + CRF (Huang et al., 2015)	94.46	90.10	-
Flair Embedding (Akbik et al., 2018)	96.72	93.09	89.3
GCDT w/ BERT (Liu et al., 2019c)	96.81	93.23	-
BERT-MRC (Li et al., 2020a)	-	93.04	91.11
HCR w/ BERT (Luo et al., 2020)	-	93.37	90.30
BERT-Biaffine Model (Yu et al., 2020)	-	**93.5**	91.3
LUA	**97.02**	93.47	**92.01**

Table 2: Experiment results on syntactic chunking and NER.

2.3 Training Criterion

We adopt hinge loss as the training criterion. Given the predicted segmentation $\widehat{\mathbf{y}}$ and the ground truth segmentation $\mathbf{y}^*$, we have

$$\mathcal{J} = \max\big(0, 1 - f(\mathbf{y}^*) + f(\widehat{\mathbf{y}})\big). \qquad (9)$$

Cross-entropy is also a widely used loss function. However, our experiments show its results are slightly worse than those of hinge loss.

3 Experiments

We have performed a series of studies to show the effectiveness and efficiency of LUA.

3.1 Settings

We use the same neural networks configurations for all the datasets. The dimensions of scoring layers are 512. L2 regularization and dropout ratio are respectively set as 1×10^{-6} and 0.2 to avoid overfitting. The batch size is 8. The above setting is obtained by grid search. We utilize Adam (Kingma and Ba, 2014) to optimize our model. Our models all run on NVIDIA Tesla P100 GPUs. At test time, we convert the predicted segments into IOB format and use conlleval script[2] to compute the F1 score. Besides, the improvements of our model over the baselines are statistically significant under t-test with a reject probability small than 0.05%.

3.2 Results on Chinese POS Tagging

Chinese POS tagging jointly segments a Chinese character sequence and assigns a POS tag to every segments. We use Chinese Treebank 5.0 (CTB5), CTB6, CTB9 (Xue et al., 2005), and the Chinese section of Universal Dependencies 1.4 (UD1) (Nivre et al., 2016). We follow the same train/dev/test splits and formats of these datasets as in Shao et al. (2017).

Table 1 diagrams the experiment results. The performances of all the baselines are copied from Meng et al. (2019); Tian et al. (2020). LUA has notably outperformed prior methods and yielded state-of-the-art results on all the datasets. Our improvements of F1 scores over baselines are 0.22% on PTB5, 0.62% on CTB6, and 0.16% on CTB9, and 0.54% on UD1. BERT Tagging is a strong baseline, and LUA outperforms it by 0.78%, 0.62%, 2.86%, and 1.30% on these datasets.

3.3 Results on Chunking and NER

Syntactic chunking aims to recognize the phrases related to syntactic category for a sentence. We use CoNLL-2000 dataset (Sang and Buchholz, 2000). The original dataset contains a training set and a test set. We take 1000 cases from the training set by uniform sampling and treat them as a development

[2] https://www.clips.uantwerpen.be/conll2000/chunking/conlleval.txt.

Method	CTB9	UD1
LUA	**94.93**	**96.02**
w/o BERT, w/ Bi-LSTM	92.18	90.53
w/o composition score $s^c_{i,j}$	94.65	95.67
w/o hinge loss, w/ cross-entropy	94.81	95.86

Table 3: Ablation experiments on two datasets.

set. NER recognizes the key phrases in a sentence and assigns a label to every extracted phrase. We use CoNLL-2003 dataset (Sang and De Meulder, 2003) and OntoNotes 5.0 dataset (Pradhan et al., 2013). We follow the same format and partition of them as in Li et al. (2020a).

The results are shown in Table 2. We follow the F1 scores of baselines reported in Akbik et al. (2018); Li et al. (2020a); Luo et al. (2020); Yu et al. (2020). Besides, Luo et al. (2020) find the evaluation method of GCDT is non-standard, and thus we re-test its performance on CoNLL-2000 with its open-source code[3]. LUA has achieved state-of-the-art results on CoNLL-2000 and OntoNotes 5.0, and performed competitively on CoNLL-2003. Our F1 scores outnumber those of baselines by 0.22% on CoNLL-2000 and 0.78% on OntoNotes 5.0. LUA only underperforms BERT-Biaffine Model by 0.03% on CoNLL-2003. Compared with a strong baseline, Flair Embedding, LUA outperforms it by 0.31% on CoNLL-2000, 0.41% on CoNLL-2003, and 3.03% on OntoNotes 5.0.

3.4 Ablation Studies

Table 3 shows our studies to examine the impacts of some parts of LUA.

Effect of the Sentence Encoder. We use BERT to exploit the knowledge from unlabeled corpora. Replacing it with LSTM (Hochreiter and Schmidhuber, 1997) sharply reduces our F1 scores by 2.98% on CTB9 and 6.06% on UD1.

Effect of the Scoring Model. LUA scores the labels and the spans independently (or only label scores $s^l_{i,j,t}, (i,j,t) \in \mathbf{y}$ are left in Eq. (3)). This improves the results of our model by 0.30% on CTB9 and 0.37% on UD1.

Effect of the Loss Function. We find that using hinge loss leads to slightly better results than cross-entropy. Their performance gaps are 0.13% and 0.17% on the two datasets.

Method	CTB9	CoNLL-2000
LUA	94.93	97.02
w/ Label Correlations	**95.08**	**97.16**

Table 4: The comparisons of whether to incorporate the label correlations or not.

Method	Time Complexity	Running Time
BERT	$\mathcal{O}(n\|\mathcal{L}\|)$	4m39s
BERT + CRF	$\mathcal{O}(n\|\mathcal{L}\|^2)$	6m41s
LUA	$\mathcal{O}(n^2\|\mathcal{L}\|)$	6m17s

Table 5: Comparing different methods in terms of running time on CoNLL-2000.

3.5 Capturing Label Correlations

Following CRF and Semi-Markov CRF, we parameterize a matrix $\mathbf{W}^d \in \mathbb{R}^{|\mathcal{L}| \times \mathcal{L}}$ to model the label dependencies among segments. Specifically, we add a term, $\sum_{1 \le k \le m} \mathbf{W}^d_{t_{k-1},t_k}$, into the scoring function, Eq. (3). The results are shown in Table 4. Explicitly capturing label correlations slightly improves our F1 scores by 0.16% on CTB9 and 0.14% on CoNLL-2000.

3.6 Running Time Analysis

Table 5 shows the comparison between baselines and LUA on efficiency. The last two columns are respectively the theoretical time complexity and the one-epoch training time cost of every method. Inspired by Zhang et al. (2020b), through parallel matrix computation on GPU, the time complexity of BERT can be reduced to $\mathcal{O}(1)$, and those of others can also be optimized to $\mathcal{O}(n)$.

We can see that LUA is a relatively fast model. For example, its time cost for training is less than that of BERT + CRF, a strong baseline, by 6.37%. We conclude that LUA is both effective and efficient for practical usage.

4 Related Work

The traditional method to sequence segmentation converts it into a sequence labeling tasks with IOB tagging scheme. This method is simple and effective, which has inspired a lot of well-performed models (Huang et al., 2015; Lample et al., 2016; Li et al., 2020b). For example, Akbik et al. (2018) present Flair Embeddings that pre-trains character embedding in a large corpus and directly use it, instead of word representation, to encode a sentence. Luo et al. (2020) use hierarchical contextualized

[3] https://github.com/Adaxry/GCDT.

representations to incorporate both sentence-level and document-level information.

Recently, span-based models have received much attention. They treat a span, instead of a token, as the basic unit for labeling. For instance, Yu et al. (2020); Li et al. (2020c) rank all the spans in terms of the scores predicted by a biaffine model (Dozat and Manning, 2016). Span-based models also emerge in other fields. Stern et al. (2017) integrate LSTM-minus feature into constituent parsing models.

5 Conclusion

This work presents a unified span-based model, LUA, for neural sequence segmentation. Given a natural language sentence, we use BERT to encode it and apply DP to extract the segmentation candidate with the maximum score. Extensive experiments have been conducted on 3 tasks across 7 datasets. LUA has established new state-of-the-art results on 6 of them. We have gained further improvements through explicitly modeling the label dependencies among segments.

LUA is now adopted as an NER option in our online text understanding system, Texsmart (Zhang et al., 2020a; Liu et al., 2021).

Acknowledgments

We thank anonymous reviewers for their kind and constructive suggestions.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649.

Galen Andrew. 2006. A hybrid Markov/semi-Markov conditional random field for sequence segmentation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 465–472, Sydney, Australia. Association for Computational Linguistics.

Richard Bellman. 1966. Dynamic programming. *Science*, 153(3731):34–37.

Deng Cai and Hai Zhao. 2016. Neural word segmentation learning for Chinese. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–420, Berlin, Germany. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*.

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

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lingpeng Kong, Chris Dyer, and Noah A Smith. 2016. Segmental recurrent neural networks. In *International Conference on Learning Representations*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020a. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Yangming Li, lemao liu, and Shuming Shi. 2021. Empirical analysis of unlabeled entity problem in named entity recognition. In *International Conference on Learning Representations*.

Yangming Li, Han Li, Kaisheng Yao, and Xiaolong Li. 2020b. Handling rare entities for neural sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6441–6451.

Yangming Li, Lemao Liu, and Shuming Shi. 2020c. Empirical analysis of unlabeled entity problem in named entity recognition. *arXiv preprint arXiv:2012.05426*.

Lemao Liu, Haisong Zhang, Haiyun Jiang, Yangming Li, Enbo Zhao, Kun Xu, Linfeng Song, Suncong Zheng, Botong Zhou, Jianchen Zhu, et al. 2021. Texsmart: A system for enhanced natural language understanding.

Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. 2019a. Towards improving neural named entity recognition with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5301–5307.

Yijia Liu, Wanxiang Che, Yuxuan Wang, Bo Zheng, Bing Qin, and Ting Liu. 2019b. Deep contextualized word embeddings for universal dependency parsing. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(1):1–17.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019c. GCDT: A global context enhanced deep transition architecture for sequence labeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2431–2441, Florence, Italy. Association for Computational Linguistics.

Ying Luo, Fengshun Xiao, and Hai Zhao. 2020. Hierarchical contextualized representation for named entity recognition. In *AAAI*, pages 8441–8448.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for chinese character representations. In *Advances in Neural Information Processing Systems*, pages 2746–2757.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.

Erik F Sang and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task: Chunking. *arXiv preprint cs/0009008*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Sunita Sarawagi and William W Cohen. 2005. Semi-markov conditional random fields for information extraction. In *Advances in neural information processing systems*, pages 1185–1192.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*.

Yan Shao, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. 2017. Character-based joint segmentation and POS tagging for Chinese using bidirectional RNN-CRF. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 173–183, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.

Yuanhe Tian, Yan Song, and Fei Xia. 2020. Joint Chinese word segmentation and part-of-speech tagging via multi-channel attention of character n-grams. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2073–2084, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, 11(2):207.

Zhixiu Ye and Zhen-Hua Ling. 2018. Hybrid semi-Markov CRF for neural sequence labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 235–240, Melbourne, Australia. Association for Computational Linguistics.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6470–6476, Online. Association for Computational Linguistics.

Feifei Zhai, Saloni Potdar, Bing Xiang, and Bowen Zhou. 2017. Neural models for sequence chunking. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Haisong Zhang, Lemao Liu, Haiyun Jiang, Yangming Li, Enbo Zhao, Kun Xu, Linfeng Song, Suncong Zheng, Botong Zhou, Jianchen Zhu, et al. 2020a. Texsmart: A text understanding system for fine-grained ner and enhanced semantic analysis. *arXiv preprint arXiv:2012.15639*.

Yu Zhang, Zhenghua Li, and Min Zhang. 2020b. Efficient second-order TreeCRF for neural dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3295–3305, Online. Association for Computational Linguistics.

Yue Zhang and Jie Yang. 2018. Chinese NER using lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1564, Melbourne, Australia. Association for Computational Linguistics.