A Neural Network based Translation Constrained Reranking Model for Chinese Dependency Parsing

Miaohong Chen, Baobao Chang and Yang Liu

Key Laboratory of Computational Linguistics, Ministry of Education
School of Electronics Engineering and Computer Science, Peking University
Collaborative Innovation Center for Language Ability, Xuzhou 221009 China
miaohong-chen@foxmail.com, chbb@pku.edu.cn, cs-ly@pku.edu.cn

Abstract. Bilingual dependency parsing aims to improve parsing performance with the help of bilingual information. While previous work have shown improvements on either or both sides, most of them mainly focus on designing complicated features and rely on golden translations during training and testing. In this paper, we propose a simple yet effective translation constrained reranking model to improve Chinese dependency parsing. The reranking model is trained using a max-margin neural network without any manually designed features. Instead of using golden translations for training and testing, we relax the restrictions and use sentences generated by a machine translation system, which dramatically extends the scope of our model. Experiments on the translated portion of the Chinese Treebank show that our method outperforms the state-of-the-art monolingual Graph/Transition-based parsers by a large margin (UAS).

Keywords: Bilingual Dependency Parsing, Reranking, Neural Network, Machine Translation

1 Introduction

Dependency parsing is a crucial task of natural language processing (NLP) and has been intensively explored during the last decades. Dominant dependency parsing methods mainly falls into two categories: transition-based [1, 2] and graph-based [3]. Both methods demonstrated relatively high accuracies on parsing English texts. However, the performance of existing Chinese dependency parsers is still considerably inferior compared to their English counterpart, due to the limited size of annotated Treebank and the morphology-poor characteristics of the language.

To improve the accuracy of Chinese dependency parsing and reduce the performance gap, we propose a neural network based translation constrained reranking model for Chinese dependency parsing in this paper. Our motivation for this model is twofold. 1) Using deep learning methods to avoid complicated and tricky feature engineering. Instead of manually designing various linguistics-motivated features, we simply use distributed representations as the input of the reranking model. Embeddings of words and pos tags as well as the interactions between them are automatically learning through the neural network. 2) Enhancing Chinese dependency parsing performance with the help

of automatically generated English translations. This is motivated by the fact that Chinese and English are not always ambiguous in the same way. When a Chinese sentence is syntactically ambiguous, its English translation might not be the case. Specifically, we train the reranking model with dependency annotated Chinese sentences in the training set as well as their translations generated by a machine translation system. Since the English translations are automatically generated, the model does not rely on golden aligned sentences for training or testing any more, which makes our method much more applicable.

We conduct our experiments on the commonly used translated portion of Chinese Treebank. Experimental results show that the reranking model improves two state-of-the-art dependency parsing systems (transition-based and graph-based) by a large margin. To show the impact of the translation noise on the performance, we also train a model with golden translations. The comparison shows that only slight performance difference is observed and the reranking model is robust enough to work with the rather noisy translations.

The rest of this paper is organized as follows: Section 2 introduces some most related work. Then the model is detailed in Section 3. Section 4 shows the experimental results we conducted on the translated portion of Chinese Treebank. We conclude this paper in Section 5.

2 Related Work

Bilingual constraints have been mostly investigated by previous work to biparsing task. [4] combined three statistical models into a unified bilingual parser that jointly searches for the best English parse, Korean parse and word alignments. They showed that bilingual constraints can be leveraged to transfer parse quality from a resource-rich language to resource-impoverished one. [5] presented a log-linear model over triples of source trees, target trees and node-to-node tree alignments between them. They also showed that parsing with joint models on bilingual texts improves performance on either or both sides. However, since both [4] and [5] aimed to improve the parsing performance of source and target language jointly, they required not only bilingual texts but also syntactic trees on both sides, which are hard to obtain.

[6] proposed a bilingually constrained monolingual shift-reduce parsing model. They introduced several bilingual features based on word alignment information to resolve what they called shift-reduce conflicts. [7] proposed a dependency parsing method that uses bilingual subtree constraints. They used a subtree list collected from large scale automatically parsed data on the target side as additional features for the source side dependency parser. Although [6] and [7]'s work focus on improving the parsing performance of one language at a time, they still need bilingually aligned sentences for both training and testing, which makes their methods not applicable in common cases where golden translations are not available. [**?**] proposed a method to improve the accuracy of parsing bilingual texts (bitexts) with the help of statistical machine translation (SMT) systems. But their method needs a monolingual parser on the target side and very large auto-parsed sentences are used.

There are mainly two differences between our work and most previous methods. Firstly, our model avoids complicated manually designed features since it is based on deep learning methods, while most previous work focus on designing complicated features. Secondly, most previous work rely heavily on bilingually aligned sentences, which is hard to obtain. Thus their application scenarios are limited to bilingual processing. We see this problem from the opposite point of view and use automatically translated sentences during training and testing. This makes our model much more applicable.

3 Reranking Model

3.1 Scoring Candidate Trees with Neural Network

Suppose we have a Chinese sentence c, its corresponding English translation e and a set of word-to-word alignments A where $a = (i, a_i) \in A$ means that the ith Chinese word in c (c_i) is aligned to the a_ith English word in e (e_{a_i}). Now we have a candidate dependency tree t of sentence c, what we want to do is scoring this tree with all the information we have. First, we score the entire tree t as follows:

$$s_t(t|c, e, A) = \sum_{i=1}^{|c|} s_a(Context(c_i)) \tag{1}$$

where $|c|$ denotes the length of sentence c. This means that we have a score at each position of c according to its contextual information $Context(c_i)$, and then sum them up to be the score of the tree.

Now we introduce how to obtain $Context(c_i)$ given $c,\ e,\ t$ and A. We take the following information into consideration:

- $c[i - ws : i + ws]$: c_i's context words in c, ws is the parameter for window size which we set to 2.
- $pos[i - ws : i + ws]$: POS tags of c_i's context words in c.
- $c_{p_i},\ c_{lc_i}, c_{rc_i}$: corresponds to c_i's parent, left-most child and right-most child in tree t separately.
- $e[a_i - ws : a_i + ws]$: c_i's aligned English word and its context words in e.

We use dense feature embeddings for words and POS tags [18] in $Context(c_i)$. Once we get $Context(c_i)$, we concatenate all the word and POS tag embeddings together to form a feature vector x_i. Then we take x_i as the input of a neural network which has one hidden layer:

$$f(x_i) = W_2[\sigma(W_1 x_i + b_1)] + b_2 \tag{2}$$

where σ is an element-wise activation function. $\theta = (W_1, b_1, W_2, b_2)$ is the parameters of the neural network.

As we can see, there are only unigrams of words and POS tags in $Context(c_i)$, no complicated features are designed at all. The feature embeddings and interactions of words and POS tags are automatically learned by the neural network.

For simplicity, we let $s(t, \theta)$ denotes $s_t(t|c, e, A)$ as in Equation 1. Then we have:

$$s(t, \theta) = \sum_{i=1}^{|c|} f(x_i) \tag{3}$$

decoding becomes the problem of finding highest scoring tree among all candidate trees. Since each candidate t is generated by a monolingual parser with corresponding score $s_m(t)$, we take this into consideration, find the best candidate tree t^* as follows:

$$t^* = \arg \max_{t \in T(c)} \lambda s(t, \beta) + (1 - \beta) s_m(t) \tag{4}$$

where $T(c)$ is a candidate set generated by monolingual parser and β is the weighting coefficient.

3.2 Max-Margin Training with AdaGrad

Since the neural network model is reranking-oriented, our training goal is that the highest scoring tree will always be the golden tree $\hat{t}$. And its score will be larger up to a margin than any other candidate trees. Then the max-margin criteria requires that for each tree t in the candidate set $T(c)$, the following inequality holds:

$$s(\hat{t}, \theta) \geq s(t, \theta) + \Delta(\hat{t}, t) \tag{5}$$

where $\Delta(\hat{t}, t)$ is the margin loss denotes the discrepancy between trees. It is measured by counting the number of words whose parent is different:

$$\Delta(\hat{t}, t) = \kappa \sum_{i=1}^{|c|} I(p_{\hat{t},i} \neq p_{t,i}) \tag{6}$$

where $\kappa = 0.1$ is a discount parameter, $I(\cdot)$ is an indicator function and $p_{t,i}$ is the parent of c_i in tree t. This leads to our final regularized objective function:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} r_i(\theta) + \frac{\lambda}{2} ||\theta||_2^2, \text{where}$$
$$r_i(\theta) = \max_{t \in T(c)} (s(t, \theta) + \Delta(\hat{t}_i, t)) - s(\hat{t}_i, \theta) \tag{7}$$

As this objective is minimized, the score of the golden tree $\hat{t}$ increases and the score of the highest scoring incorrect candidate tree decreases. Following [8] and [9], we use the subgradient method and diagonal variant of AdaGrad [10] with minibatches to optimize this function.

4 Experiments

4.1 Setup

Bilingual Data Following [6] and [7], we conduct our experiments on the translated portion of Chinese Treebank (CTB) [11, 12], articles 1-325, which have golden English

translations. We use Penn2Malt[1] to convert the data into dependency trees. We also use the same split as in [6], which is shown in Table 1. Table 1 also shows the number of bilingual pairs we extracted from the bilingual articles. Note that not all sentence pairs can be included, since many of them are not one-to-one aligned at sentence level.

Monolingual Baselines In order to generate candidate trees for the reranking model, we train two state-of-the-art baseline parsers using the rest articles of CTB: a second-order graph-based parser trained using MSTParser[2] [3], and a self implemented transition-based parser, whose features templates are used following [13] . The best k parse trees generated by the baseline parsers are treated as the candidate set $T(c)$.

Alignment and MT System Since bilingual features are extracted through word alignment, we train a word alignment model with the GIZA++[3] implementation of IBM4 using approximately 0.8M bilingual sentence pairs, which do not include the CTB data. We also remove the notoriously bad links in $\{$a, an, the$\} \times \{$的(DE), 了(LE)$\}$ following the work of [6]. To generate English translations for Chinese sentences, we use the same 0.8M sentence pairs to train a phrase-based translation model with Moses [14] and tune the parameter using minimum error rate training with other 3k sentence pairs.

Hyperparameters We use the development set to select hyperparameters for the reranking model. Finally, the reranking models are trained for 15 iterations, the hidden layer's size $h = 200$, word and POS tag embedding size $n = 50$, candidate set size $k = 12$, regularization weight $\lambda = 0.0001$, and initial learning rate $= 0.05$. For the weighting coefficient in Equation 4, we set $\beta = 0.2$ for graph-based parser and $\beta = 0.8$ for transition-based parser.

Initialization We initialize Chinese/English word embeddings with embeddings pre-trained on Chinese/English Gigaword [15, 16] using Word2Vec[4]. Chinese POS tag embeddings and the neural network's weight matrices (W_1, W_2) are randomly initialized using uniform distribution $[0.2, 0.2]$. The bias vectors b_1 and b_2 are initialized as zeros.

Table 1. Training, testing and development data from the translated portion of Chinese Treebank as in [5]

	Training	Test	Dev
CTB Articles	1-270	271-300	301-325
Bilingual Pairs	2494	263	252

4.2 Results

We test the reranking model with three different settings:

[1] http://stp.lingfil.uu.se/ nivre/research/Penn2Malt.html
[2] http://sourceforge.net/projects/mstparser/
[3] http://www.statmt.org/moses/giza/GIZA++.html
[4] http://code.google.com/p/word2vec/

– **Ours-MM:** In this setting we use English translations automatically generated by Moses for training and testing.
– **Ours-GM:** Different from "Ours-MM", "Ours-GM" uses golden English translations for training while translations for testing sentences are still generated by Moses.
– **Ours-GG:** Golden translations are used for both training and testing data.

As we can see, "Ours-MM" only depends on automatically generated translations during training and testing. Thus it will be applicable for resource-poor languages which have no translated treebank for training at all, which is often the case. In setting "Ours-GM", golden translations are only required for training. Hence it still works when golden translations of testing sentences is missing. "Ours-GG" is most similar to previous bilingual parsing models but it can only be applied to limited cases where golden translations of training and testing sentences are given.

We report unlabeled attachment score (UAS) and unlabeled exact match (UEM). Figure 1 shows the performance (UAS) of the reranking model on development set with different candidate set size k, the baseline here is graph-based. As we can see, the performance improves as k gets larger at first. When k is around 12, the performance reaches its maximum. Then it goes down along with the increase of k. Thus we finally set k to 12 for all the experiments.

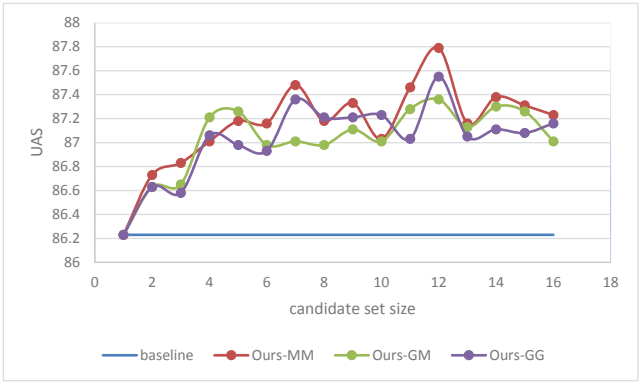

Fig. 1. Performance of the reranking model on development set with different candidate size when baseline is graph-based.

The main results are shown in Table 2. As we can see from the table, the reranking model improves the baselines by a large margin on all three settings. Compared to the two state-of-the-art baselines, our translation constrained reranking model "Ours-MM" yields an improvement of +2.06/+3.22 on UAS over the graph-based/transition-based monolingual parsers and also an improvement of +4.56/+3.42 on UEM, while "Ours-GG" gets an improvement of +2.27/+3.22 on UAS and +4.82/+3.55 on UEM. Despite the fact that it is trained and tested on automatically generated translations,

"Ours-MM" only performs slightly worse than "Ours-GG", which is trained and testing with golden translations. This indicates that the reranking model still works well even without any golden translations. Among all the three settings, "Ours-GM" performs the worst, which may caused by the inconsistency of the training (golden translation) and testing (Moses generated).

Table 2. Experimental results on test set. UAS: Unlabeled Attachment Score, UEM: Unlabeled Exact Match.

	Graph-based		**Transition-based**	
	UAS	UEM	UAS	UEM
Baseline	83.57	36.50	83.71	31.94
Ours-MM	85.63	41.06	**86.93**	35.36
Ours-GM	85.08	40.30	86.46	34.47
Ours-GG	**85.84**	**41.32**	**86.93**	**35.49**

4.3 Effect of Global and Bilingual Information

As described in Section 3.1, in our reranking model, we use two kinds of features that are not available to the baseline parses: 1) **bilingual features** extracted through word alignment, 2) **global features** such as left/right-most child in the dependency tree. In this section, we demonstrate their effects on our reranking models. Figure 2 shows the performance (UAS) of our reranking model when bilingual or global features are excluded separately. As we can see from the figure, the performances decline with the absence of bilingual or global features, but still outperform the baselines. The rerankers almost perform the best in all cases (except "MG-G") when both kinds of features are available. This indicates that, although bilingual and global features are both important to the reranking model, they provide different kinds of information and play different roles.

4.4 Effect of Word Embedding Initialization

All the experiments showed in the above sections use word embeddings pre-trained on Gigaword with Word2Vec. In this section we further analyze the influence of using pre-trained word embeddings as for initialization. In order to show the comparison, we randomly initialize all word embedding using uniform distribution$[0.2, 0.2]$. From Figure 3 we can see that, using pre-trained word embeddings tends to produces better results than using randomly initialized word embeddings. But the differences are barely obvious, and the rerankers still get comparable accuracies without the help of pre-trained word embeddings.

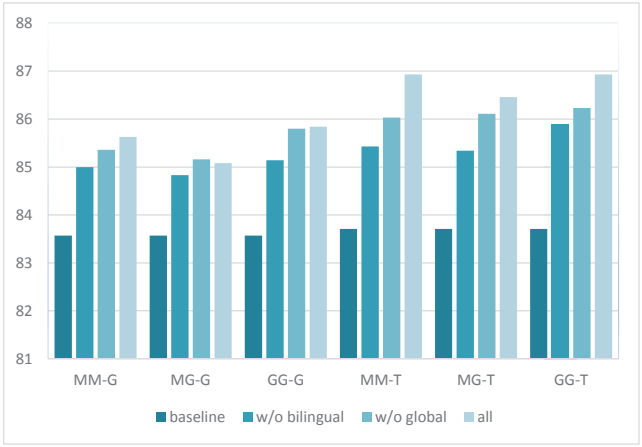

Fig. 2. The effects of global and bilingual information. "w/o bilingual" represents the reranking model without bilingual feature, "w/o global" represents the reranking model without global features. "all" means both kinds of features are used in the reranking model. "–G" means that the baseline is graph-based and "–T" means transition-based. For instance, "MM-G" represents the reranking model with setting "Ours-MM" and the baseline is graphs-based.

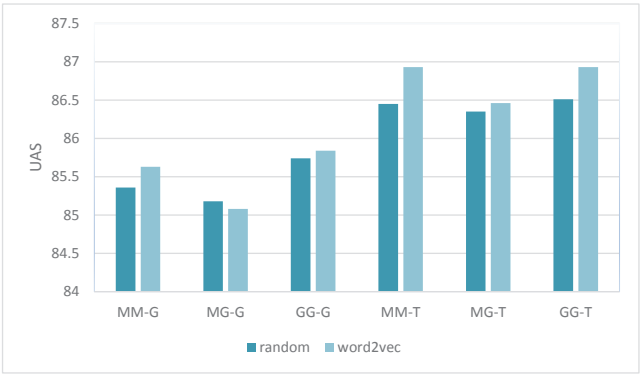

Fig. 3. Effect of word embedding initialization. Systems share the same meaning as in Figure 2.

4.5 Comparison with Previous Work

As most previous work focused on biparsing task or assumed to work with bilingual texts, a direct comparison with them is hard to conduct. Table 3 shows the performance of some similar systems. [6] and [7] reported bilingual dependency parsing results on the translated portion of CTB. We notice that there are still some obvious differences among us. First, the performances of the monolingual baseline parsers are barely the same. Second, we all use different extra resources. Third, our method is reranking-based. Hence it is not proper to directly compare our work with theirs. But as a rough indicator, we just consider the absolute improvement of UAS over baseline. As show in Table 3, [6] and [7] separately get +0.6 and +2.93 improvement. For our models, "Ours-GG" is most similar to their settings and gets +2.27/+3.22 improvement over graph/transition based baselines.

Table 3. Comparison with similar work (UAS).

System	Baseline	Final	Improvement
Huang09	85.70	86.30	+0.60
Chen10	87.2	90.13	+2.93
Ours-GG-G	83.57	85.84	+2.27
Ours-GG-T	83.71	86.93	**+3.22**

5 Conclusion and Discussion

This paper presents a neural network based translation constrained reranking model for Chinese dependency parsing. Distributed feature representations free us from complicated and tricky feature designing. We use machine automatically generated rather than golden English translations for both training and testing. This makes our model be applicable in much broader scenarios and enables it to continuously benefit from the improvement of machine translation techniques. Experimental results show that the reranking model outperforms both the graph-based and transition-based monolingual models by a large margin. Due to its simplicity, our method can be easily applied to any other resource-poor languages.

Acknowledgments

This work is supported by National Key Basic Research Program of China(2014CB340504) and National Natural Science Foundation of China(61273318).

References

1. Hiroyasu Yamada and YujiMatsumoto. 2003. Statistical dependency parsing analysis with support vector machines. *In proceedings of the Eighth International Workshop on Parsing technologies (IWPT)*.

2. Joakim Nirve. 2003. An efficient algorithm for projective dependency parsing. *In proceedings of the Eighth International Workshop on Parsing technologies (IWPT).*
3. Ryan McDonald, Koby Grammer, and Fernando Peereira. 2005. Online large-margin training of dependency parsers. *Proceedings of the 43rd annual meeting on association for computational linguistics. Association for Computational Linguistics*, page 91-98.
4. David A. Smith and Doah A. Smith. 2004. Bilingual parsing with factored estimation: Using english to parse Korean. *In proceedings of EMNLP*.
5. David Burkett and Dan Klein. 2008. Two languages are better than one (for syntactic paring). *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics*, page 877-886.
6. Liang Huang, Wenbin Jiang, and Qun Liu. 2009. Bilingually-constrained (monolingual) shift-reduce parsing. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3. Association for Computational Linguistics*, page 1222-1231.
7. Wenliang Chen, Jun'ichi Kazama, and Kentaro Torisawa. 2010. Bitext dependency parsing with bilingual subtree constraints. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics*, page 21-29.
8. Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013. Parsing with compositional vector grammars. *In proceedings of ACL.*
9. Wenzhe Pei, Tao Ge, and Baobao Chang. 2014. Max-margin tensor neural network for Chinese word segmentation. *In proceedings of ACL.*
10. John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, page 2121-2159.
11. Nianwen Xue, Fu-Dong Chiou, and Martha Palmer. 2002. Building a large-scale annotated Chinese coupus. *Proceedings of the 19th international conference on Computational linguistics-Volume 1. Association for Computational Linguistics*, page 1-8.
12. Ann Bies, Martha Palmer, Justin Mott and Colin Warner. 2007. English Chinese translation treebank v1.0. LDC2007T02.
13. Liang Huang and Kenji Sagae. 2010. Dynamic programming for linear-time incremental parsing. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics*, page 1077-1086.
14. Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions. Association for Computational Linguistics*, page: 177-180.
15. David Graff and Ke Chen. 2003. Chinese Gigaword. LDC2003T09.
16. David Graff and Christopher Cieri. 2003. English Gigaword. LDC2003T05.
17. Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *The Journal of Machine Learning Research.*
18. Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 740-750.