Cross-Lingual Dependency Parsing by POS-Guided Word Reordering

Lu Liu[1,2], **Yi Zhou**[1,2], **Jianhan Xu**[1,2],
Xiaoqing Zheng[1,2], **Kai-Wei Chang**[3], **Xuanjing Huang**[1,2]
[1]School of Computer Science, Fudan University, Shanghai, China
[2]Shanghai Key Laboratory of Intelligent Information Processing
[3]University of California, Los Angeles, USA
{luliu19, yizhou17, xujh16, zhengxq}@fudan.edu.cn
kwchang@cs.ucla.edu, xjhuang@fudan.edu.cn

Abstract

We propose a novel approach to cross-lingual dependency parsing based on word reordering. The words in each sentence of a source language corpus are rearranged to meet the word order in a target language under the guidance of a part-of-speech based language model (LM). To obtain the highest reordering score under the LM, a population-based optimization algorithm and its genetic operators are designed to deal with the combinatorial nature of such word reordering. A parser trained on the reordered corpus then can be used to parse sentences in the target language. We demonstrate through extensive experimentation that our approach achieves better or comparable results across 25 target languages (1.73% increase in average), and outperforms a baseline by a significant margin on the languages that are greatly different from the source one. For example, when transferring the English parser to Hindi and Latin, our approach outperforms the baseline by 15.3% and 6.7% respectively.

1 Introduction

The rise of machine learning (ML) methods and the availability of treebanks (Buchholz and Marsi, 2006) for a wide variety of languages have led to a rapid increase in research on data-driven dependency parsing (McDonald and Pereira, 2006; Nivre, 2008; Kiperwasser and Goldberg, 2016). However, the performance of dependency parsers heavily relies on the size of corpus. Due to the great cost and difficulty of acquiring sufficient training data, ML-based methods cannot be trivially applied to low-resource languages.

Cross-lingual transfer is a promising approach to tackle the lack of sufficient data. The idea is to train a cross-lingual model that transfers knowledge learned in one or multiple high-resource source languages to target ones. This approach has been successfully applied in various tasks, including part-of-speech (POS) tagging (Kim et al., 2017), dependency parsing (McDonald et al., 2011), named entity recognition (Xie et al., 2018), entity linking (Sil et al., 2018), question answering (Joty et al., 2017), and coreference resolution (Kundu et al., 2018).

A key challenge for cross-lingual parsing is the difficulty to handle word order difference between source and target languages, which often causes a significant drop in performance (Rasooli and Collins, 2017; Ahmad et al., 2019). Inspired by the idea that POS sequences often reflect the syntactic structure of a language, we propose CURSOR (**C**ross ling**U**al pa**RS**ing by w**O**rd **R**eordering) to overcome the word order difference issue in cross-lingual transfer. Specifically, we assume we have a treebank in the source language and annotated POS corpus in the target language[1]. We first train a POS-based language model on a corpus in the target language. Then, we reorder words in each sentence on the source corpus based on the POS-based language model to create pseudo sentences with target word order. The resulting reordered treebank can be used to train a cross-lingual parser with multi-lingual word embeddings.

We formalize word reordering as a combinatorial optimization problem to find the permutation with the highest probability estimated by a POS-based language model. However, it is computationally difficult to obtain the optimal word order. To find a near-optimal result, we develop a population-based optimization algorithm. The algorithm is initialized with a population of feasible solutions and iteratively produces new generations by specially designed genetic operators. At each iteration, better solutions are generated by applying selection, crossover, and mutation subroutines to individuals in the previous iteration.

Our contributions are summarized as follows:

[1]It is much easier to annotate POS than a treebank.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2938–2948
November 16 - 20, 2020.

(i) We propose a novel cross-lingual parsing approach, called CURSOR, to overcome the word order difference issue in cross-lingual transfer by POS-guided word reordering. We formalize word reordering as a combinatorial optimization problem and develop a population-based optimization algorithm to find a near-optimal reordering result.

(ii) Extensive experimentation with different neural network architectures and two dominant parsing paradigms (graph-based and transition-based) shows that our approach achieves an increase of 1.73% in average UAS, if English is taken as the source language and the performance is evaluated on other 25 target languages. Specifically, for the RNN-Graph model, our approach gains an increase of 2.5% in average UAS, and the improvement rises to 4.12% by the combination of our data augmentation and ensemble method.

(iii) Our approach performs exceptionally well when the target languages are quite different from the source one in their word orders. For example, when transferring the English RNN-Graph parser to Hindi and Latin, our approach outperforms a baseline by 15.3% and 6.7%, respectively.

2 Related Work

Many efforts (Zeman and Resnik, 2008; Cohen et al., 2011; Rosa and Žabokrtskỳ, 2015) have been devoted to cross-lingual dependency parsing via transfer learning, in which manually annotated corpora are no longer required for low-resource languages. One of the challenges is the word orders in source and target languages might be different (e.g., some languages are prepositional and some are postpositional). Various studies have been dedicated to addressing this issue (Naseem et al., 2012; Zhang and Barzilay, 2015; Wang and Eisner, 2017).

In particular, some studies proposed to bypass word order issue by selecting source languages that have similar word orders to the target language (Naseem et al., 2012; Rosa and Žabokrtskỳ, 2015). Good source languages can be selected by measuring the similarity of POS sequences between the source and target languages (Agic, 2017), querying the information stored in topological databases (Deri and Knight, 2016), and formalizing such selection as a ranking problem (Lin et al., 2019).

Treebank translation (Tiedemann et al., 2014; Tiedemann and Agić, 2016) tackles this problem by transforming an annotated source treebank to instances with target language grammar through machine translation. However, this method may suffer from imperfect word alignment between two languages. Zhang et al. (2019) proposed to perform such syntactic transfer by code mixing in which only the confident words in a source treebank will be transformed.

Another interesting solution to cross-lingual transfer is an annotation projection (Hwa et al., 2005; Ganchev et al., 2009; Ma and Xia, 2014). In this approach, source-side sentences of a parallel corpus are parsed by the parser trained on the source treebank, then the source dependencies are projected onto the target sentences using the results of word alignments. However, the resulting treebank could be highly noisy because the source dependency trees are constructed automatically and cannot be taken as ground truth. Lacroix et al. (2016) considered removing not well-aligned sentences to obtain high-quality data.

Täckström et al. (2013) trained a parser on multiple source languages instead of a single one. Ponti et al. (2018) proposed a typologically driven method to reduce anisomorphism. Ahmad et al. (2019) designed an order-free model to extract the order features from the source language. Meng et al. (2019) embraced the linguistic knowledge of target languages to guide the inference. Some researchers also exploit lexical features to enhance the parsing models. Cross-lingual word clusters (Täckström et al., 2012), word embeddings (Guo et al., 2015, 2016; Ammar et al., 2016), and dictionaries (Durrett et al., 2012; Rasooli and Collins, 2017) are used as the features to better transfer linguistic knowledge among different languages.

Our work is in line with a recently proposed solution, namely treebank reordering (Wang and Eisner, 2016, 2018; Rasooli and Collins, 2019), which aims to rearrange the word order in source sentences to make them more similar to the target one. Wang and Eisner (2018) proposed to permute the constituents of an existing dependency treebank to make its surface POS statistics approximately match those of the target language. However, they used POS bigrams to measure the surface closeness between two languages, which is unable to capture global information. Rasooli and Collins (2019) proposed two different syntactic reordering methods, one is based on the dominant dependency direction in the target language, the other learns a reordering classifier, but both methods rely on parallel corpus.

In this study, we explore the feasibility of utiliz-

ing a POS-based neural language model to guide treebank reordering. Our approach does not require any parallel corpus, and can be applied to a pair of source and target languages as long as their POS tags are available. We designed a population-based optimization algorithm to deal with the combinatorial nature of word reordering. This algorithm is able to find the close-to-optimal results of reordering, which yields a new state-of-the-art for cross-lingual parsing in various languages.

3 Approach

In this section, we first formalize the word reordering as a combinatorial optimization problem, and then present our method to solve the problem.

3.1 Problem Definition

Given a sentence $x = \{x_1, x_2, ..., x_n\}$ in the source dataset $\mathcal{S}$, we aim to permute the words in the sentence to mimic the order in the target language. To measure the goodness of a permutation, we train a POS-based language model $p_\mathcal{T}$ on the target corpus $\mathcal{T}$ using a multi-layer LSTM. The log-likelihood of a sentence under $p_\mathcal{T}$ can be formulated as follows:

$$p_\mathcal{T}(x) = \prod_{i=1}^{n} p_\mathcal{T}(x_i \| x_{<i}). \quad (1)$$

The objective is to find one permutation x^* so that the reordered sentence will achieve a high probability estimated by the language model:

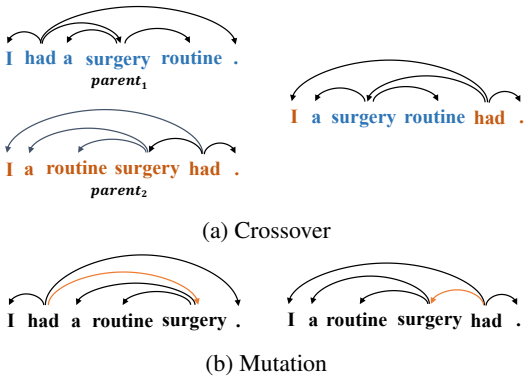
$$x^* = \arg \max_{x' \in R(x)} p_\mathcal{T}(x'), \quad (2)$$

where $R(x)$ is a set of all possible permutations of the words in x. In theory, the number of the feasible candidates is $n!$, while most of the permutations may be radically different from the original sentence and break the meaning. To avoid that, we apply a syntactic constraint when generating $R(x)$: *a sub-sequence that forms a constituent in the original sentence should still be a sub-sequence after reordering, while the inner order of words in the sub-sequence may change*.

3.2 Population-based Optimization

Finding the optimal x^* in Equation (2) can be reduced to a well-known travelling salesperson problem[2], which is NP-hard. Therefore, the optimal reordering is computationally difficult to obtain, and we design a genetic algorithm to find near-optimal results instead.

[2]If we consider words as cities, the best word order as the shortest possible route.



(a) Crossover

(b) Mutation

Figure 1: Example mutation and crossover operators.

Genetic algorithm is a heuristic search method inspired by the process of natural selection, which iteratively evolves a population of candidate solutions towards better ones. The population of each iteration is called a generation. The algorithm starts by executing *initialization* operator to create the initial generation. At each generation, the *fitness* of every individual in the population is evaluated, and individuals with higher fitness score have more chance to breed the next generation by applying *selection* operator. The next generation is produced through a combination of two genetic operators: *crossover* and *mutation*. The *crossover* operator combines the genetic information of two parents to generate new offspring, while the *mutation* operator introduces diversity into the sampled population. Genetic algorithms are known to perform well in solving combinatorial optimization problems (Anderson and Ferris, 1994; Mühlenbein, 1989) and are suitable for the word reordering problem.

In order to meet the syntactic constraint, we design the *crossover* and *mutation* operators at the subtree level, which means whenever a word is moved to some other place, the subtree of it should be moved at the same time. We describe each components of the proposed genetic algorithm below:

Fitness: The fitness score of an individual is defined by its log-likelihood in the target language model as Equation (1).

Selection: In a generation, "fitter" solutions are more likely to be selected for breeding the next generation. We normalize the fitness score of sentences in the generation and use it as the probability that each sentence may be selected randomly.

Crossover: We use the example shown in Figure 1a to better describe the crossover operator. Given two parents $parent_1$ and $parent_2$ chosen randomly by the selection operator, we then ran-

Algorithm 1 Genetic algorithm-based reordering

Input: $\mathcal{S}$: source treebank; N_g: the number of generations; N_p: the number of populations; α: mutation probability
Output: reordered treebank $\mathcal{S}'$

for $x_{orig} \in \mathcal{S}$ **do**
  for $i = 1, \cdots, N_p$ **do**
    $P_i^0 = \textit{Mutation}(x_{orig})$
  end for
  for $g = 1, \cdots, N_g$ **do**
    $P^g = P^{g-1}$
    for $i = 1, \cdots, N_p$ **do**
      $F_i^{g-1} = p_\mathcal{T}(P_i^{g-1})$
    end for
    $p_{selection} = \text{Normalize}(F^{g-1})$
    for $i = 1, \cdots, N_p$ in population **do**
      Sample $parent_1$ from P^{g-1} with $p_{selection}$
      Sample $parent_2$ from P^{g-1} with $p_{selection}$
      $child = \textit{Crossover}(parent1, parent2)$
      if UniformSampling(0, 1) $<\alpha$ **then**
        $child = \textit{Mutation}(child)$
      end if
      Add $child$ to P^g
    end for
    $P^g =$ top-N_p elements in P^g with largest fitness
  end for
  $x^* = \arg\max_{x \in P^g} p_\mathcal{T}(x)$
  Add x^* to $\mathcal{S}'$
end for

domly pick a word ("surgery" in the example) as the crossover point. We copy the entire inside tree ("a surgery routine" in $parent_1$) and then merge it with the remaining words as the order occurred in $parent_2$ to produce an offspring sentence.

Mutation: We move a child node (along with its subtree) from one side of its head node to the opposite side. An example of mutation is shown in Figure 1b, we first randomly select a pair of words ("had" $\rightarrow$ "surgery"), and then move the word "surgery" and its subtree to the left side of the head word "had".

Initialization: We repeatedly apply the mutation operator (discussed above) to the original sentence to generate an initial generation.

The overall algorithm is listed in Algorithm 1. For each sentence in $\mathcal{S}$, the descendant with the highest fitness score is added to the reordered treebank $\mathcal{S}'$. After reordering the corpus, a parser trained on $\mathcal{S}'$ can be used to analyse the target language since the instances in $\mathcal{S}'$ are conformed with the grammar of the target language.

4 Experiments

We evaluate CURSOR by transferring four different parsing models trained on English corpus to 30 target languages. We first introduce the experimental setup, then discuss the results as well as in-depth analysis, and finally, we propose a combined approach to further improve the performance.

4.1 Setup

Data We conduct experiments on Universal Dependencies (UD) Treebanks (v2.2) (Nivre et al., 2018), in which 31 different languages (one as the source and others as target languages) are selected for evaluation. The number of tokens is more than 100K for each selected language. We take English as the source language and 30 other languages as target ones. 5 target languages are used to tune the hyperparameters and remaining 25 languages are held out for final evaluation.

Parsing Models We evaluate CURSOR with four different parsing models described by Ahmad et al. (2019): SelfAtt-Graph, RNN-Graph, SelfAtt-Stack, and RNN-Stack. These models are built upon two encoders (SelfAtt/RNN) as well as two decoders (Graph/Stack). RNN encoder uses bidirectional LSTMs while SelfAtt encoder uses a transformer (Vaswani et al., 2017) instead. Graph decoder utilizes a deep biaffine attentional scorer proposed by Dozat and Manning (2017), and Stack decoder is a top-down transition-based decoder proposed by Ma et al. (2018).

Lexicalized Features Following (Ahmad et al., 2019), all the parsing models take words as well as their gold POS tags as input. We also leverage pre-trained multilingual embeddings from FastText (Bojanowski et al., 2017) that project the word embeddings from different languages into the same space using an offline transformation method (Smith et al., 2017; Conneau et al., 2018).

Training Details For fair comparison, we use the same hyper-parameter settings and the training strategy as Ahmad et al. (2019) to train the parsing models. Each POS-based language model for word reordering is trained on the training set of a corresponding target language, in which the POS tag dimension is set to 50 (as the same as that in the parsing models), the hidden size $h \in \{50, 100\}$ and the number of layers $l \in \{1, 2, 3\}$ are tuned on the development sets of 5 non-held-out languages. In Algorithm 1, we introduce three new hyperparameters of N_p, N_g, α , and thier values are tuned from a few choices: $N_p \in \{5, 10, 20\}, N_g \in \{5, 10, 20\}, \alpha \in \{0.5, 0.8, 1.0\}$. On the five non-held-out target languages, the best performance is obtained with the

Lang	SelfAtt-Graph		RNN-Graph		SelfAtt-Stack		RNN-Stack	
	Baseline	CURSOR	Baseline	CURSOR	Baseline	CURSOR	Baseline	CURSOR
en	90.4/88.4	-	90.4/88.3	-	90.2/88.1	-	91.8/89.9	-
sl	68.2/56.5	68.7/56.5 ↑	66.3/54.6	68.9/56.9 ↑	66.6/54.6	66.6/54.1 ↑	67.8/55.7	70.2/54.7 ↑
cs	63.1/53.8	65.6/55.2 ↑	61.9/52.8	65.1/55.8 ↑	61.3/51.9	63.9/53.3 ↑	62.3/52.3	64.8/54.2 ↑
ro	65.1/54.1	67.6/56.2 ↑	63.2/52.1	67.4/56.8 ↑	62.5/51.5	64.5/53.0 ↑	61.0/49.8	65.9/54.5 ↑
zh*	42.5/25.1	39.8/24.1	41.5/24.3	40.3/24.1	40.6/23.3	37.2/20.4	40.9/23.5	39.9/21.9
ja*	28.2/20.9	41.6/32.5 ↑	18.4/12.0	37.6/29.9 ↑	20.7/13.2	38.9/30.7 ↑	15.2/9.3	40.7/31.9 ↑
Average	53.4/42.1	**56.7/44.9** ↑	50.3/39.2	**55.9/44.7** ↑	50.3/38.9	**54.2/42.3** ↑	49.4/38.1	**56.3/44.0** ↑
no	80.8/72.8	77.5/69.7	80.7/72.8	77.9/70.5	80.3/72.1	76.4/68.6	81.8/73.3	78.7/70.7
sv	81.0/73.2	78.2/70.5	81.2/73.5	79.2/71.6	80.6/72.8	77.8/70.0	82.6/74.3	80.1/71.8
fr	77.9/72.8	79.2/74.2 ↑	78.4/73.5	79.9/74.9 ↑	76.8/71.8	78.1/72.8 ↑	75.5/70.5	79.3/74.2 ↑
pt	76.6/67.8	76.7/67.0 ↑	76.5/68.0	77.3/68.2 ↑	75.4/66.7	75.3/65.4	74.6/66.1	76.8/67.4 ↑
da	76.6/67.9	75.5/67.1	77.4/68.8	76.7/68.2	76.4/67.5	74.7/66.1	78.2/68.8	75.7/66.7
es	74.5/66.4	74.1/65.9	74.9/66.9	75.2/66.7 ↑	73.2/65.1	72.9/64.9	73.1/64.8	75.1/66.8 ↑
it	80.8/75.8	81.0/75.6 ↑	81.1/76.2	81.4/76.3 ↑	79.1/74.2	79.2/73.9 ↑	80.4/75.3	81.2/76.2 ↑
hr	61.9/52.9	64.0/52.9 ↑	60.1/50.7	65.2/54.9 ↑	60.6/51.1	62.0/50.8 ↑	60.8/51.1	62.0/51.4 ↑
ca	73.8/65.1	74.2/65.4 ↑	74.2/65.6	74.6/65.9 ↑	72.4/63.7	72.8/63.9 ↑	72.0/63.0	73.7/65.1 ↑
pl	74.6/62.2	79.2/66.7 ↑	71.9/58.6	78.6/66.3 ↑	73.5/60.5	78.5/65.4 ↑	72.1/59.8	78.5/65.5 ↑
uk	60.1/52.3	62.1/53.2 ↑	58.5/51.1	60.2/52.0 ↑	57.4/49.7	56.4/48.0	59.7/51.9	59.8/50.9 ↑
nl	68.6/60.3	69.1/61.5 ↑	67.9/60.1	70.2/62.8 ↑	67.9/59.5	68.2/60.7 ↑	69.6/61.6	70.4/63.3 ↑
bg	79.4/68.2	78.4/67.1	78.1/66.7	79.3/67.6 ↑	78.2/67.0	76.2/64.8	78.8/67.6	79.1/67.8 ↑
ru	60.6/51.6	62.3/52.7 ↑	60.0/50.8	62.8/53.5 ↑	59.4/50.3	60.1/50.0 ↑	60.9/52.0	59.9/50.5
de	71.3/61.6	75.9/67.1 ↑	69.5/59.3	76.5/67.8 ↑	69.9/60.1	73.7/65.1 ↑	69.6/59.6	76.7/68.1 ↑
he	55.3/48.0	56.3/48.9 ↑	54.6/46.9	57.2/50.6 ↑	53.2/45.7	53.8/46.6 ↑	54.9/41.0	55.0/44.7 ↑
sk	66.7/58.2	69.9/59.7 ↑	65.4/57.0	68.5/59.6 ↑	65.3/56.7	67.9/57.3 ↑	66.6/57.5	69.7/59.2 ↑
id	49.2/43.5	54.8/47.4 ↑	47.1/42.1	52.1/46.4 ↑	47.3/41.7	53.2/45.4 ↑	46.8/41.3	53.1/46.2 ↑
lv	70.8/49.3	66.3/46.7	71.4/49.6	68.9/49.1	69.0/47.8	63.7/44.4	70.6/48.5	69.0/48.6
fi	66.3/48.7	63.9/47.3	66.4/48.7	64.7/47.8	64.8/47.5	60.8/43.9	66.3/48.3	64.3/47.1
et	65.7/44.9	65.1/46.0	65.3/44.4	64.9/46.0	64.1/43.3	61.4/43.2	64.3/43.5	63.5/44.7
ar	38.1/28.0	42.9/32.9 ↑	33.0/25.5	38.2/31.2 ↑	32.6/23.7	38.4/29.4 ↑	32.9/25.0	38.8/30.5 ↑
la	48.0/35.2	52.9/38.2 ↑	46.0/33.9	52.7/38.8 ↑	45.5/33.2	51.0/36.2 ↑	43.9/31.3	52.6/37.6 ↑
ko	34.5/16.4	36.2/19.3 ↑	33.7/15.4	37.3/19.9 ↑	32.8/15.0	33.3/17.4 ↑	33.1/14.3	35.6/18.4 ↑
hi	35.5/26.5	45.1/34.4 ↑	29.3/21.4	44.6/34.9 ↑	31.4/23.1	41.8/32.3 ↑	25.9/18.1	44.1/34.4 ↑
Average	65.1/54.8	**66.4/55.9** ↑	64.1/53.9	**66.6/56.4** ↑	63.5/53.2	**64.3/53.9** ↑	63.8/53.1	**66.1/55.5** ↑

Table 1: Cross-lingual transfer performance (UAS%/LAS%, punctuation excluded) on the test sets. We use English as the source language and the first five languages to tune the hyperparameters. The languages listed are sorted in ascending order by their distances to English as reported by Ahmad et al. (2019). We use '$*$' to indicate the results of delexicalized models.

setting of $h = 100, l = 2, N_p = 10, N_g = 10$ and $\alpha = 0.5$.

Methods for Comparison We mainly compare CURSOR to the models described by Ahmad et al. (2019), denoted as "Baseline", which is different from CURSOR in that the words of the sentences from source languages are not reordered. We also compare CURSOR to two models proposed by Wang and Eisner (2018) and Meng et al. (2019), respectively denoted as MiniDiver and LagraRelax. MiniDiver is also based on word reordering, which reorders the words of the source sentences to minimize the difference in POS sequence distribution between the source and the target languages. LagraRelax solves the word order difference problem by using a Lagrangian relaxation to force the constraints derived from corpus-statistics in the inference time, which yields a significant improvement in transfer parsing. Different external resources are used by these approaches. MiniDiver assumes that the target POS corpus is available like CURSOR, while LagraRelax utilizes World Atlas of Language Structures (WALS) (Dryer and Haspelmath, 2013) linguistic features.

4.2 Results

We report in Table 1 the results of Baseline and CURSOR on the test sets for 30 different languages. Those languages are sorted in ascending order by their typology distances to English as reported by Ahmad et al. (2019). Following their recommendation, we use delexicalized models where only POS tags are used as inputs for two target languages of Chinese (zh) and Japanese (ja) since their word embeddings were found to be not well aligned with those of the others.

As we can see from Table 1, comparing to the baseline, the cross-lingual transfer performances

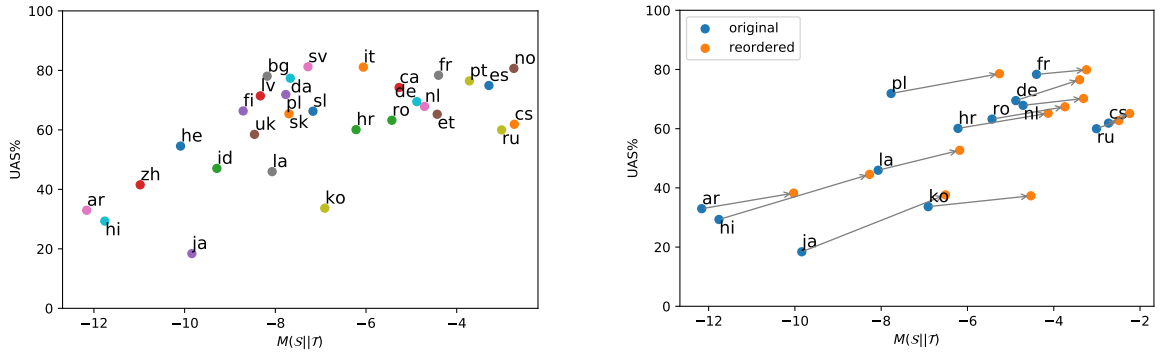


Figure 2: Transfer parsing performance versus similarity between languages. (a) shows the correlation between the transfer performance and the similarity of source and target languages in their word orders. (b) demonstrates that by increasing the similarity in their word orders our method can substantially improve the transfer performance.

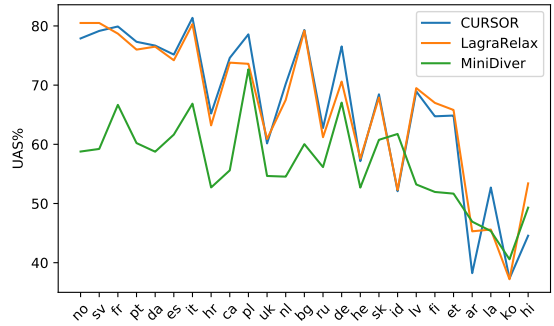


Figure 3: Comparison with the competitors. CURSOR outperforms MiniDiver in most languages, and achieves slightly better results than LagraRelax.

are all improved with four different parsing models trained on the corpora after the word reordering. The models using RNN encoder benefit more than others probably because they are more sensitive to the word orders than those using SelfAtt encoder. RNN-Graph model enhanced by our treebank reordering achieved the best average UAS of 66.6%, which beats the baseline by 2.5%. The improvements are exceptionally significant for those languages whose word orders are quite different from English, such as Hindi (hi) and Latin (la).

We report in Figure 3 the results of comparing our approach to other competitors. The results of CURSOR are those achieved by the model based on RNN-Graph architecture. For MiniDiver, we use the code released by Wang and Eisner (2018) to reorder source treebanks, then train an RNN-Graph parser on the reordered treebank. The results of LargraRelax are excerpted from Meng et al. (2019). It shows that CURSOR performs better than MiniDiver in almost all languages, which demonstrates that the POS-based neural language model can lead to better results of word reordering than the bigram language model. Besides, CURSOR achieves slightly better results than LagraRelax (the average UAS of CURSOR is 66.6%, while that of LagraRelax is 66.3%). However, our reordering method can be applied to both the graph-based and transition-based parsing paradigms, while LagraRelax can only be used for the graph-based parsing. Furthermore, the performance of CURSOR can be further improved to 68.21% by the combination of our data augmentation and ensemble method (see Section 4.4).

Although all the experimental results reported so far take English as the source language, our approach can be applied to the case where any language is chosen as the source language without any additional effort. We also run experiments in which Hebrew (he) is taken as the source language. Experimental results with four different parsing models show that CURSOR can consistently improve the average UAS across 30 target languages by 4.23%, 6.48%, 2.91%, and 5.52% respectively.

4.3 Analysis

In this section, we study the relationship between the cross-lingual transfer parsing performances and the similarities of the source and target languages, and how the difference in arc directionality and arc distance impact on the performance.

4.3.1 Performance versus Similarity between Languages

We here first validate our hypothesis that "if two languages have higher similarity, the transfer performance will be better". Then, we demonstrate

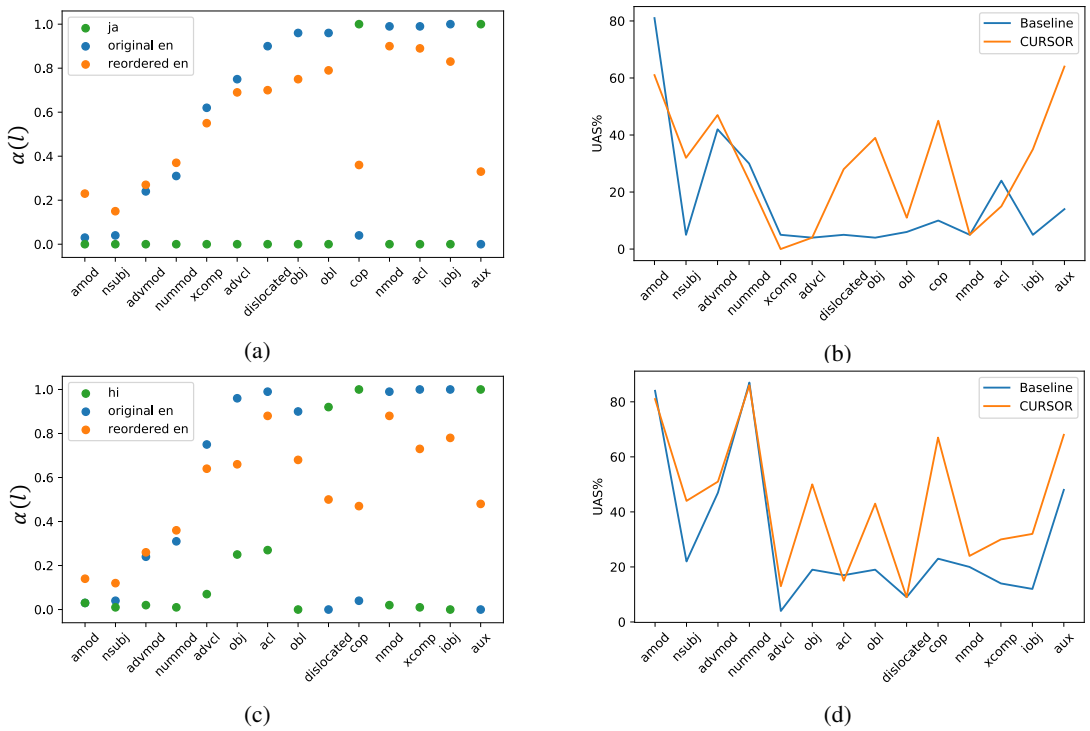


Figure 4: Analysis in Japanese (ja) and Hindi (hi), the values of α are calculated on the training sets. **(a)** and **(c)** show that the differences in the directionality between source and target corpus can be reduced by our word reordering method. **(b)** and **(d)** show that large differences will lead to poor transfer performance, and CURSOR can benefit from the reduced differences.

that our word reordering method can make two different languages "closer" in their typology distance, which usually leads to an improvement in the cross-lingual transfer.

We define a metric M to measure how a source language $\mathcal{S}$ is similar to a target one $\mathcal{T}$ with the help of the POS-based language model $p_{\mathcal{T}}$ as follows:

$$M(\mathcal{S}\|\mathcal{T}) = \frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}}\frac{1}{|x|}\log p_{\mathcal{T}}(x) \quad (3)$$

We show the correlation between the transfer performance and the similarity of source and target languages in Figure 2a, and found that they are correlated in general, especially when the value of M is less than -8. Figure 2b shows that after reordering $\mathcal{S}$, its similarity to $\mathcal{T}$ increases, and the corresponding cross-lingual parsing performance will improve. Particularly, target languages with greater differences to the source one in their word order will benefit more from our reordering method.

4.3.2 Performance versus Difference in Arc Directionality

We will show that given a specific arc label, the transfer performance is significantly affected by the difference in the directionality (Wang and Eisner, 2017) of the source and target languages, and demonstrate that CURSOR can reduce such difference thus improving the performance.

Given a label l, we define the directionality $\alpha(l) \in [0, 1]$ as the probability that a modifier is at the right side of its head. For the label l, the difference of directionality between the source (English) and target language $\mathcal{T}$ can be calculated as:

$$\delta_{\mathcal{T}}(l) = |\alpha_{en}(l) - \alpha_{\mathcal{T}}(l)| \quad (4)$$

In Figure 4, we sort the arc labels by their corresponding $\delta_{\mathcal{T}}(l)$ in ascending order. As shown in Figure 4b and 4d, large $\delta_{\mathcal{T}}(l)$ will lead to poor transfer performance. We also observe that our word reordering method can effectively reduce the difference of such directionality, which usually improves the performance of cross-lingual transfer. For example (see Figure 4a), $\delta_{ja}(cop)$ and $\delta_{ja}(aux)$ are greatly reduced after reordering. As a result, the parsing UAS of these two labels improves significantly as shown in Figure 4b (from 10.12% to 44.64% and from 13.84% to 64.09%, respectively).

4.3.3 Performance versus Arc Distance

We show in Figure 5 the parsing performances versus the arc distances for German (de). The arc distance of a modifier and its head is calculated by the

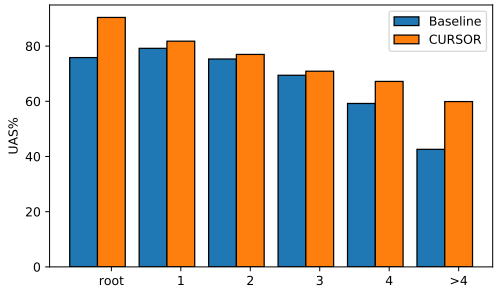


Figure 5: Performance versus arc distance. CURSOR outperforms the baseline across different arc distances.

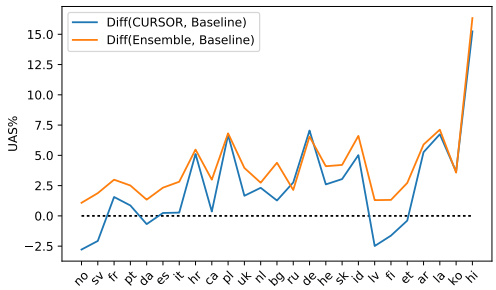


Figure 6: The performance of CURSOR can be further improved by the ensemble method in almost all target languages.

number of words staying between them. It shows that CURSOR outperforms the baseline by a significant margin in all cases. Such margin increases when the arc distance becomes longer, indicating that the model is more sensitive to the correctness of word order when making predictions on the long-distance dependencies.

4.4 Combined Approach

We here explore the feasibility of improving the cross-lingual parsing based on RNN-Graph by data augmentation and ensemble method.

4.4.1 Data Augmentation

In Algorithm 1, we only add the result of word reordering with the highest fitness score to the reordered training treebank $\mathcal{S}'$. However, the fitness scores of the top-k results are normally very close, and we try to use all these results to train the parsing model. As shown in table 2, increasing the number of the top-k word reordering results can improve the transfer parsing performance, and the highest performance is achieved when $k = 3$.

4.4.2 Model Ensemble

Although the population-based optimization can reduce the difference in word order between two languages, it may change the well-formed syntactic

Model	UAS%	LAS%
Baseline	64.09	53.90
CURSOR ($k = 1$)	66.55	56.44
CURSOR ($k = 2$)	67.04	56.86
CURSOR ($k = 3$)	67.56	57.35
CURSOR ($k = 4$)	67.49	57.30
CURSOR ($k = 1$) + Baseline	67.63	57.48
CURSOR ($k = 3$) + Baseline	**68.21**	**58.04**

Table 2: Results of RNN-Graph parser across 25 target languages in average UAS and LAS. Generally, the more number (k) of the word reordering results are used to train the model, the better the performance will be. Ensembling CURSOR ($k = 3$) with the baseline achieves the highest accuracy in both UAS and LAS.

structure of a source language. For a pair of similar languages, such change may cause a drop in the performance. We thus propose an inference-time ensemble method which combines the output of CURSOR and Baseline by:

$$\begin{aligned} w(m,h) =& \gamma_{\mathcal{T}} \cdot w_{\text{Baseline}}(m,h) \\ &+ (1-\gamma_{\mathcal{T}}) \cdot w_{\text{CURSOR}}(m,h) \\ \gamma_{\mathcal{T}} =& 0.5\times \left(1 - \frac{\max M\left(\mathcal{S}\|\cdot\right) - M\left(\mathcal{S}\|\mathcal{T}\right)}{\max M\left(\mathcal{S}\|\cdot\right) - \min M\left(\mathcal{S}\|\cdot\right)}\right) \end{aligned} \tag{5}$$

where $w(m,h)$ denotes the score that h is the head of m, $\gamma_{\mathcal{T}}$ governs the relative importance of two models, $\max M\left(\mathcal{S}\|\cdot\right)$ and $\min M\left(\mathcal{S}\|\cdot\right)$ are the highest and lowest scores computed as Equation (3) among 25 target languages. If the target language is more similar to the source one we will put more weights on Baseline.

We show in Figure 6 that the ensemble method can further improve the transfer performance of CURSOR, and outperform Baseline in all languages. Ensembling CURSOR ($k = 3$) with Baseline achieves the best performance (68.21% in UAS and 58.04% in LAS), establishing a new start-of-the-art as shown in Table 2.

5 Conclusion

We propose a treebank reordering approach for cross-lingual dependency parsing. Our approach does not require any parallel corpus and can be applied to any pair of source and target languages as long as their POS tags are available. Extensive experimentation with different network architectures across 30 languages demonstrates that our approach can substantially improve the performance of the cross-lingual parsing.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments. This work was partly supported by National Key R&D Program of China (No. 2018YFC0830900), National Science Foundation of China (No. 62076068), Shanghai Municipal Science and Technology Major Project (No. 2018SHZDZX01) and Zhangjiang Lab.

References

Željko Agic. 2017. Cross-lingual parser selection for low-resource languages. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies, 22 May, Gothenburg Sweden*, 135, pages 1–10. Linköping University Electronic Press.

Wasi Uddin Ahmad, Zhisong Zhang, Xuezhe Ma, Eduard Hovy, Kai-Wei Chang, and Nanyun Peng. 2019. On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing. In *Proceedings of NAACL-HLT*, pages 2440–2452.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

Edward J Anderson and Michael C Ferris. 1994. Genetic algorithms for combinatorial optimization: the assemble line balancing problem. *ORSA Journal on Computing*, 6(2):161–173.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Sabine Buchholz and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the tenth conference on computational natural language learning (CoNLL-X)*, pages 149–164.

Shay B Cohen, Dipanjan Das, and Noah A Smith. 2011. Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 50–61. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. *Internation Conference on Learning Representations*.

Aliya Deri and Kevin Knight. 2016. Grapheme-to-phoneme models for (almost) any language. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 399–408.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. *Internation Conference on Learning Representations*.

Matthew S Dryer and Martin Haspelmath. 2013. Wals online. leipzig: Max planck institute for evolutionary anthropology.

Greg Durrett, Adam Pauls, and Dan Klein. 2012. Syntactic transfer using a bilingual lexicon. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1–11. Association for Computational Linguistics.

Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. 2009. Dependency grammar induction via bitext projection constraints. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 369–377. Association for Computational Linguistics.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. 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)*, pages 1234–1244.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2016. A representation learning framework for multi-source transfer parsing. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, 11(3):311–325.

Shafiq Joty, Preslav Nakov, Lluís Màrquez, and Israa Jaradat. 2017. Cross-language learning with adversarial neural networks. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 226–237.

Joo-Kyung Kim, Young-Bum Kim, Ruhi Sarikaya, and Eric Fosler-Lussier. 2017. Cross-lingual transfer learning for pos tagging without cross-lingual resources. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 2832–2838.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Gourab Kundu, Avirup Sil, Radu Florian, and Wael Hamza. 2018. Neural cross-lingual coreference resolution and its application to entity linking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 395–400.

Ophélie Lacroix, Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. 2016. Frustratingly easy cross-lingual transfer for transition-based dependency parsing. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1058–1063.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, et al. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135.

Xuezhe Ma, Zecong Hu, Jingzhou Liu, Nanyun Peng, Graham Neubig, and Eduard Hovy. 2018. Stack-pointer networks for dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1403–1414.

Xuezhe Ma and Fei Xia. 2014. Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1337–1348.

Ryan McDonald and Fernando Pereira. 2006. *Discriminative learning and spanning tree algorithms for dependency parsing*. University of Pennsylvania.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the conference on empirical methods in natural language processing*, pages 62–72. Association for Computational Linguistics.

Tao Meng, Nanyun Peng, and Kai-Wei Chang. 2019. Target language-aware constrained inference for cross-lingual dependency parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1117–1128.

Heinz Mühlenbein. 1989. Parallel genetic algorithms, population genetics and combinatorial optimization. In *Workshop on Parallel Processing: Logic, Organization, and Technology*, pages 398–406. Springer.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 629–637. Association for Computational Linguistics.

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, et al. 2018. Universal dependencies 2.2. *LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University*.

Edoardo Maria Ponti, Roi Reichart, Anna Korhonen, and Ivan Vulić. 2018. Isomorphic transfer of syntactic structures in cross-lingual nlp. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1531–1542.

Mohammad Sadegh Rasooli and Michael Collins. 2017. Cross-lingual syntactic transfer with limited resources. *Transactions of the Association for Computational Linguistics*, 5:279–293.

Mohammad Sadegh Rasooli and Michael Collins. 2019. Low-resource syntactic transfer with unsupervised source reordering. 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 3845–3856.

Rudolf Rosa and Zdeněk Žabokrtskỳ. 2015. Klcpos3-a language similarity measure for delexicalized parser transfer. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 243–249.

Avirup Sil, Gourab Kundu, Radu Florian, and Wael Hamza. 2018. Neural cross-lingual entity linking. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *Internation Conference on Learning Representations*.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1061–1071. Association for Computational Linguistics.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 conference of the North American chapter*

of the association for computational linguistics: Human language technologies, pages 477–487. Association for Computational Linguistics.

Jörg Tiedemann and Zeljko Agić. 2016. Synthetic treebanking for cross-lingual dependency parsing. *Journal of Artificial Intelligence Research*, 55:209–248.

Jörg Tiedemann, Željko Agić, and Joakim Nivre. 2014. Treebank translation for cross-lingual parser induction. In *Eighteenth Conference on Computational Natural Language Learning (CoNLL 2014)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Dingquan Wang and Jason Eisner. 2016. The galactic dependencies treebanks: Getting more data by synthesizing new languages. *Transactions of the Association for Computational Linguistics*, 4:491–505.

Dingquan Wang and Jason Eisner. 2017. Fine-grained prediction of syntactic typology: Discovering latent structure with supervised learning. *Transactions of the Association for Computational Linguistics*, 5:147–161.

Dingquan Wang and Jason Eisner. 2018. Synthetic data made to order: The case of parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1325–1337.

Jiateng Xie, Zhilin Yang, Graham Neubig, Noah A Smith, and Jaime G Carbonell. 2018. Neural cross-lingual named entity recognition with minimal resources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 369–379.

Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2019. Cross-lingual dependency parsing using code-mixed treebank. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 996–1005.

Yuan Zhang and Regina Barzilay. 2015. Hierarchical low-rank tensors for multilingual transfer parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1857–1867.