A Little Perturbation Makes a Difference: Treebank Augmentation by Perturbation Improves Transfer Parsing

Ayan Das and Sudeshna Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur,
Kharagpur, WB, India
`ayan.das@cse.iitkgp.ernet.in`
`sudeshna@cse.iitkgp.ac.in`

Abstract

We present an approach for cross-lingual transfer of dependency parser so that the parser trained on a single source language can more effectively cater to diverse target languages. In this work, we show that the cross-lingual performance of the parsers can be enhanced by over-generating the source language treebank. For this, the source language treebank is augmented with its perturbed version in which controlled perturbation is introduced in the parse trees by stochastically reordering the positions of the dependents with respect to their heads while keeping the structure of the parse trees unchanged. This enables the parser to capture diverse syntactic patterns in addition to those that are found in the source language. The resulting parser is found to more effectively parse target languages with different syntactic structures. With English as the source language, our system shows an average improvement of 6.7% and 7.7% in terms of UAS and LAS over 29 target languages compared to the baseline single source parser trained using unperturbed source language treebank. This also results in significant improvement over the transfer parser proposed by Ahmad et al. (2019) that involves an "order-free" parser algorithm.

1 Introduction

Cross-lingual dependency parsing involves training a dependency parser using a treebank in one language (source language) and applying it to parse sentences in another language (target language). This can be used to develop parsers for languages for which no treebank is available.

The syntactic similarity between the source and the target languages typically plays an important role in the success of a cross-lingual transfer parser (Zeman and Resnik, 2008; Naseem et al., 2012; Søgaard, 2011). A major challenge in transfer parsing is to bridge the difference in the syntax of the source and the target languages. For example, the object usually occurs after the corresponding verb in English while the verb normally occurs at the final position in a clause in Japanese.

In order to achieve better performance of the transfer parsers, researchers have worked on the selection of syntactically similar source languages for a given target language (Søgaard, 2011; Rasooli and Collins, 2017; Wang and Eisner, 2016). Attempts have also been made towards improving the performance of the transferred parsers for a given source-target language pair by reducing the syntactic gaps between them. This is done by transforming the source language parse trees (Aufrant et al., 2016; Rasooli and Collins, 2019; Wang and Eisner, 2016, 2018; Das and Sarkar, 2019) using knowledge of the typological properties of the target language. However, these approaches are target language specific and may not give satisfactory results for multiple languages.

Recent work by Ahmad et al. (2019) proposed an "order-free" parser model that comprises of a transformer-based encoder and a graph-based decoder. They show that the self-attention mechanism of the transformer with direction independent position encoding used in their model gives rise to improved performance for transfer between distant pair of languages compared to a standard parser model that uses an RNN based encoder and stack pointer-based decoder.

In this paper, we propose a different approach for enhancing the performance of a target language independent transfer parser based on a single source language by augmenting the treebank of the source language without using any target language information. For this, we add sentences obtained by rearranging the original sentences in the treebank while keeping the parse tree of the sentence fixed. This can be construed as generating

a more general treebank which may contain sentences not conforming syntactically to the source language.

Specifically, we introduce controlled perturbation in the relative ordering of the head-dependent pairs in the source language parse trees. We stochastically alter the order of some of the head-dependent pairs in the source language sentences while keeping the head-dependent relations in the parse trees intact. This perturbation reduces the dependency of the parser on the word order in the training sentences and makes it more robust towards the variation in syntax.

We show that a stack-pointer network-based parser model (Ma et al., 2018) trained using this treebank results in improvement of the performance of the transfer parser over a baseline parsers trained on an unperturbed treebank. This parser also significantly outperforms the "order-free" parser model proposed by Ahmad et al. (2019) model by 3.8% UAS and 4.2% LAS. We also show that our target language independent approach gives a competitive performance with that of a target language specific transformation approach (Das and Sarkar, 2019).

2 Related Work

Initial work on model transfer involved training delexicalized models (Zeman and Resnik, 2008; McDonald et al., 2013) using only language independent non-lexical features such as PoS tags in the source language treebanks.

Several approaches for model transfer that incorporate lexical features in the transfer models have been reported in the literature. These include use of cross-lingual word clustering (Täckström et al., 2012), dictionary-based mapping of distributed word embeddings and projection-based bi-lingual word representations (Xiao and Guo, 2014; Guo et al., 2015; Schuster et al., 2019; Ahmad et al., 2019).

Søgaard (2011) proposed an approach for selecting training instances from source language by ranking them in terms of similarity with the target language sentences in terms of PoS tag perplexity. Naseem et al. (2012); Täckström et al. (2013); Zhang and Barzilay (2015) presented a multilingual algorithm for dependency parsing that selectively learns the aspects (some features listed in World Atlas of Language Structures (WALS) (Haspelmath, 2005)) of the source

languages relevant to the target language and ties the model parameters accordingly.

Another approach for improving the performance of cross-lingual transfer parsers is by transforming the source language parse trees to match the syntax of the target language. Aufrant et al. (2016) improves performance of the transfer parsers by transforming the source language parse trees based on the knowledge of the target language syntax derived from WALS. Das and Sarkar (2019) also proposed a similar source language treebank transformation method in which knowledge of the syntax of a target language is derived from small number annotated target language parse trees. Wang and Eisner (2016) generated synthetic treebanks by altering the word order of the source language treebanks using knowledge of the distribution of the noun and verb dependents of other real-world languages from their respective treebanks. Wang and Eisner (2018) proposed an approach for learning an optimized permutation parameter using the given source language treebank and a gold PoS tag annotated corpus in the target language. This parameter set is then applied to permute the source language parse trees to approximately match the syntax of the target language. These methods are however target language specific and may not perform well for other languages.

Bhat et al. (2017) have shown that training a parser model using *scrambled* parse trees of sentences of one domain improves performance of the parser over a parser model trained using the original treebank on test sentences of another domain. They scrambled the parse trees of sentences from newswire data and tested on conversational data. The scrambled treebank consisted either of all possible permutations of a subset of the parse trees in the original treebank, or, a fixed number of permutations of all the parse trees, where the permuted parse trees with the lowest perplexity assigned by a language model are selected.

Ahmad et al. (2019) proposed a parser algorithm that improves the quality of transfer parser independent of the target language. They have compared the performance of combinations of different encoder-decoder architectures. They consider a bidirectional LSTM based encoder (order-sensitive) and a transformer-based encoder (order-free), and, two types of decoders, stack-pointer based (order-sensitive) and a biaffine graph-based

(order-free) and have shown that overall best cross-lingual performance of a parser across several target languages can be achieved using the combination of transformer-based encoder and graph-based decoder model. This system is expected to be agnostic to the word order of the source sentence and thus work effectively for a variety of target languages.

Multi-source transfer (McDonald et al., 2011; Rosa and Zabokrtsky, 2015) parsing approaches combine treebanks of multiple source languages to train cross-lingual transfer parsing models.

3 Perturbation of Source Language Parse Trees

3.1 Parse Tree Structure based Perturbation

We now discuss the details of our stochastic perturbation algorithm. We call this perturbation scheme as **PTSPert**. In order to introduce variation in word order in the source language parse trees, we apply perturbation on each parse tree in the treebank which randomly changes the relative ordering of some head-dependent word pairs in the sentence. For each node in the parse tree, we classify each of its dependents as either *pre-dependent* or *post-dependent* based on whether it appears before or after its head word in the sentence. During perturbation, we convert a pre-dependent to post-dependent and vice versa with some probability. The probability of altering the relative position of a dependent with respect to its head word in a sentence is referred to as *perturbation probability* (P).

The PTSPert algorithm takes the original source language parse tree T_s as input and returns the perturbed sentence as output.

For each node n in the parse tree T_s, we maintain four lists: *pre-modifiers list ($initpre_n$), post-modifiers list ($initpost_n$), final pre-modifiers list ($finalpre_n$)* and *final post-modifiers list ($finalpost_n$)*. The *pre-modifiers list* and *post-modifiers list* contain the pre-modifiers and post-modifiers of the node in the same sequence as they appear in the original sentence. The *final pre-modifiers list* and *final post-modifiers list* are initially *empty*.

The steps of the PTSPert algorithm are as follows;

1. Traverse the words in the sentence from left to right. For each word in the sentence; let w be the node in T_s corresponding to the word.

(a) Traverse $initpre_w$ from left to right. For each dependent in the list;
   i. With probability P, append the dependent to $finalpost_w$
   ii. With probability $1 - P$, append the dependent to $finalpre_w$.
(b) Traverse $initpost_w$ from left to right. For each dependent in the list;
   i. With probability P, append the dependent to $finalpre_w$
   ii. With probability $1 - P$, append the dependent to $finalpost_w$.

The in-order traversal of the perturbed tree T_s based on the *finalpre*s and *finalpost*s of the nodes return the sentence with the new word-order.
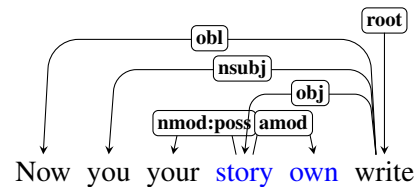
Example In Figure 2 we present the perturbed version of the sentence "Now you write your own story". The words whose positions have changed after perturbation are shown in red and blue. The final sentence after perturbation is "Now you your story own write".
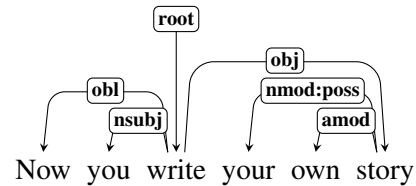
(a) English sentence before perturbation
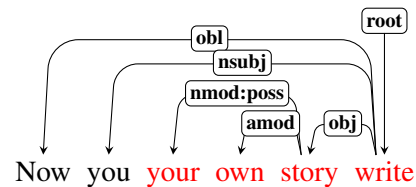
(b) Perturbation at "write"

(c) Perturbation at "story"

Figure 1: Perturbation on an English sentence.

After perturbation, the subtree with the word "story" as the head becomes a pre-dependent of the word "write". (Figure 1b) and the adjective "own" of "story" is converted to a post modifier. (Figure 1c)

3.2 Alternative Perturbation Models

Perturbation or introduction of noise in data is not new in natural language processing. It has been used to train a system to reconstruct the original sentence from its corrupted version (Dai and Le, 2015; Hill et al., 2016).

Artetxe et al. (2018) used a perturbation approach in unsupervised machine translation to learn the internal structure of a language and to reduce the dependence on the word order of the sentences to address the differences in the source and target languages. This was done by training an encoder-decoder system to recover the original sentence from its corrupted version given as input.

In this perturbation method, given a sentence of length N, N/k random swaps are made between the contiguous words, where k is a integer parameter. Artetxe et al. (2018) used $k = 2$. We call this perturbation approach **SwapPert**.

Some target language specific perturbation approaches extensively used in dependency parsing are discussed in Section 2.

4 Data and Parser Model

Data We carried out our experiments using treebanks of 29 languages from the UD v2.2 treebanks. We used the language-independent UD UPOS tags and dependency relations. We have used the acronyms of the language names in the rest of the paper. The full names of the languages are listed in Appendix A.1.

Word Embeddings We have used 300-dimensional fasttext (Bojanowski et al., 2017) pre-trained word embeddings for each language. The cross-lingual word embeddings were obtained by projecting the monolingual embeddings for all the languages into the space of the English language (Smith et al., 2017).

4.1 Parser

We have experimented with parser models with two types of encoder-decoder based parser models. The models are as follows;

- **RS:** Stack-pointer-based parser model (Ma et al., 2018) with BiLSTM RNN (Schuster and Paliwal, 1997; Hochreiter and Schmidhuber, 1997) based encoder and stack-pointer-based decoder model (Ma et al., 2018).

- **TG:** Transformer (Vaswani et al., 2017) based encoder with relative position repre-

sentation (Shaw et al., 2018) and biaffine graph based decoder (Dozat and Manning, 2017). This encoder-decoder combination is due to Ahmad et al. (2019).

For our experiments, we have used the implementations of the parsers and the corresponding hyperparameter settings by Ahmad et al. (2019).[1]

5 Experiments and Results

We carried out the experiments corresponding to the different perturbation approaches under the following settings.

SwapPert Given a sentence of length N, for N/k perturbations, we have carried out separate experiments with $k = 2$ and $k = 10$. The stack-pointer-based parser model (Ma et al., 2018) (RS) was trained for this perturbation.

PTSPertRS This refers to the stack-pointer-based parser model (Ma et al., 2018) (RS) parser model trained using a source language treebank augmented with its versions perturbed by PTSPert. We experimented with different perturbation probability values ($P \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$).

STATtrans This refers to the stack-pointer-based parser model (Ma et al., 2018)(RS) parser model trained using source language treebank transformed using statistical knowledge of target language syntax derived from samples of 20 target language parse trees (Das and Sarkar, 2019). For each target language, we randomly sampled 20 parse trees from combined training and development sets. We trained separate models specific to each target language.

5.1 Baselines

RSUnpert This refers to the stack-pointer-based parser model (Ma et al., 2018)(RS) trained on unperturbed source language treebank.

TGUnpert This is the parser model comprising of a transformer-based encoder and a graph-based decoder (Ahmad et al., 2019) (TG) trained on unperturbed source language treebank.

All our experiments were repeated 5 times and we report the average result in this paper.

---

[1]The implementation was obtained from `https://github.com/uclanlp/CrossLingualDepParser`

TL	RSUnpert	TGUnpert	PTSPertRS (P=0.2)	STATtrans
en	**91.2/89.3**	90.3/88.4	91.2/89.3	89.9/87.6
no	**81.8/73.6**	80.3/72.2	81.1/73.5	79.7/72.4
sv	**82.3/74.3**	80.9/73.1	82.2/74.3	82.1/74.7
fr	76.1/70.7	78.6/73.4	**80.7/76.1**	79.9/75.0
pt	74.6/65.8	77.0/68.1	**79.1/69.8**	78.9/69.4
da	77.6/68.5	77.1/68.4	**78.1/69.3**	76.9/68.2
es	73.4/65.1	74.9/66.9	**77.4/69.1**	78.1/70.3
it	80.4/75.4	80.6/75.4	**83.9/79.4**	84.6/79.4
hr	61.1/51.5	62.4/52.5	**63.5/53.1**	66.7/56.8
ca	72.1/63.1	73.9/65.3	**76.0/66.9**	76.2/66.5
pl	72.5/60.1	75.4/62.8	**79.4/66.7**	80.9/69.2
uk	60.0/52.0	59.8/51.6	**62.6/53.3**	63.3/55.3
sl	68.0/56.4	68.6/56.6	**68.8/56.9**	69.5/57.4
bg	79.6/68.0	**80.1/68.7**	79.6/68.6	80.5/69.4
ru	61.2/52.2	61.4/51.9	**62.8/53.1**	64.4/55.3
de	69.2/59.3	72.0/62.1	**77.1/68.5**	78.6/69.7
he	56.4/45.0	55.9/46.9	**56.6/48.2**	58.2/50.6
cs	62.6/52.9	63.3/54.0	**64.8/54.3**	63.3/54.1
ro	61.9/50.6	66.3/55.1	**67.8/56.3**	69.9/59.3
sk	66.6/57.5	67.5/58.9	**69.4/59.2**	69.6/60.3
id	46.6/41.2	49.5/43.6	**55.0/47.8**	57.8/50.2
fi	**66.4/49.0**	66.6/48.6	66.0/48.6	66.5/49.2
et	64.6/44.1	**66.0/45.9**	64.1/44.9	67.1/47.6
zh*	41.3/24.2	40.3/24.0	**41.9/24.1**	44.8/28.4
ar	33.7/25.8	38.2/28.2	**43.3/33.6**	44.2/35.3
la	44.7/32.1	48.0/35.2	**51.3/37.1**	54.4/39.8
ko	33.6/14.4	**34.2/16.7**	33.9/16.4	39.3/21.5
hi	26.6/18.4	35.0/26.5	**45.0/35.9**	69.9/55.9
ja	15.0/9.3	27.2/19.4	**38.8/30.7**	60.8/46.8
Avg	62.1/52.0	63.8/53.8	**66.3/56.1**	68.8/58.5

Table 1: UAS%/LAS% corresponding to different perturbation methods on the target languages. '*' indicates the results corresponding to the delexicalized models. The underlined entries indicate the cases where *STATtrans* performs better than *PTSPertRS*

5.2 Results with English as Source Language

Evaluation Metric We report the results of our experiments in terms of unlabeled attachment score (UAS) and labelled attachment score (LAS) excluding punctuation and symbols.

In Table 1 we report the performance of the RSUnpert, TGUnpert and PTSPertRS ($P = 0.2$) and STATtrans on 29 target languages with English as the source language. The target languages are ordered according to their typological similarity with the English language based on the metric given by Ahmad et al. (2019). For the Chinese (zh) and Japanese (ja) languages, we report the results of the delexicalized transfer parsers for a fair comparison with the baseline. The best performance for PTSPertRS was achieved at P=0.2.

We observe that perturbation results in an overall improvement in the performance of the cross-lingual transfer parsers. Our proposed approach (*PTSPertRS*) performs better than the RSUnpert baseline parser in case of 24 out of 29 target languages. It improves cross-lingual performance of the transferred parser by **6.69%** and **7.74%** in terms UAS and LAS respectively. *PTSPertRS* also performs better than *TGUnpert* in case of 25 out of 29 target languages and improves average scores by 3.8%UAS and 4.2%LAS.

We also observe that although the *PTSPertRS* is a target language independent approach it gives better performance than *STATtrans* in case of 7 languages out of 29 target language. Furthermore, the parser model with transformer-based-encoder and graph-based-decoder (TG) trained using the treebank perturbed by PTSPert also performs better than *TGUnpert* and *RSUnpert*. However, it performs slightly worse than *PTSPertRS*.

In Table 2 we summarize the performance of the different approaches discussed in this paper in terms of UAS% and LAS% averaged over all 29 target languages with English as the source language. We observe that for different values of perturbation probability, PTSPertRS outperforms RSUnpert, TGUnpert and SwapPert. We also observe that SwapPert performs slightly better than RSUnpert for $k = 10$.

Consider the following German sentence (DE) and its English gloss (EN).

DE: *"Ich kann diese Tauch schule jeden empfehlen"*

EN: *I recommend this driving school to everyone.*

This is parsed by a transfer parser trained on English. The words and relations indicated in red show the errors by *RSUnpert* parser. The error is possibly because the verb *empfehlen* occurs at the end and after the object (Tauchschule), whereas the verbs occur before the objects in most English sentences. It is observed that the *PTSPert* parser correctly parses the sentence. This may have been made possible by perturbation of the source treebank resulting in instances of verb-final occurrences in the augmented treebank.

5.2.1 Dependency Relation-wise Analysis

In Table 3 we compare the labelled accuracies of PTSPertRS ($P = 0.2$) with RSUnpert and TGUnpert corresponding to 18 most frequent dependency relations averaged across all the 29 target languages. We observe that PTSPertRS performs better than RSUnpert and TGUnpert in terms of the *case, nmod, nsubj, amod, obl, advmod, acl, obj, aux, mark* and *cc* relations.

However, PTSPertRS performs worse than either RSUnpert or TGUnpert in terms of the *advcl,*

	RSUnpert	TGUnpert	PTSPert					SwapPert		STATtrans
			0.1	0.2	0.3	0.4	0.5	$N/2$	$N/10$	
UAS	62.1	63.8	65.8	**66.2**	66.1	66.1	64.8	59.7	63.6	68.7
LAS	52.0	53.8	55.7	**56.1**	55.6	55.6	54.4	49.3	53.7	58.4

Table 2: Comparison of average performance of different transfer approaches.

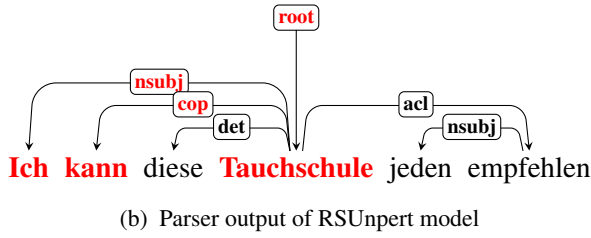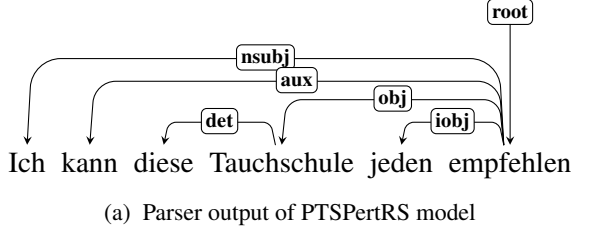(a) Parser output of PTSPertRS model

(b) Parser output of RSUnpert model

Figure 2: Parses of a German sentence.

det, cop, nummod, compound, xcomp and *flat* relations. We note that the group of words related by *compound, fixed* and *flat* relations are usually arranged sequentially in a sentence and the dependents with *appos* relation always follow their respective heads. Thus perturbation with respect to these relations negatively affects the performance of the parsers. Furthermore, TGUnpert performs better than the PTSPertRS model in terms of the *det, nummod, cop, iobj* and *appos* relations. We observed that the dependents with *cop, nummod* and *det* relations appears before their head words in English. In case of the languages in which the copulas, determiners and numeric modifiers predominantly appears after their head words, the PTSPertRS shows an overall improvement of 17.25%, 4.14% and 50.0% respectively over the TGUnpert model. However, it loses out in terms of average accuracy in case of the other languages by 4.32%, 1.06% and 4.28% respectively. Since these relations appear before their respective heads in majority of the languages which includes English, the overall accuracy is less in terms of these relations.
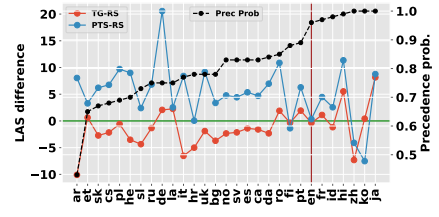
For a dependency relation, we call probability of the dependents occurring before their heads in a language as the *precedence probability* of that relation in that language. The precedence probability of a relation in a language is measured by the
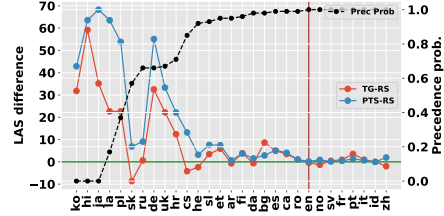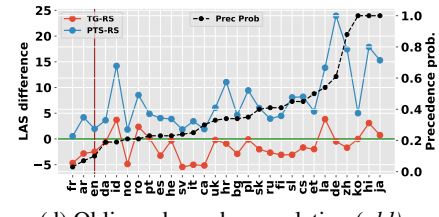

(a) Auxiliary dependency relation (*aux*)
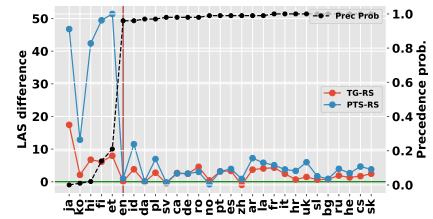
(b) Case dependency relation (*case*)

(c) Subject dependency relation (*nsubj*)

(d) Oblique dependency relation (*obl*)

Figure 3: The blue and the red lines indicate the gain in LAS by PTSPertRS and TGUnpert over RSUnpert respectively. The black line indicates the precedence probabilities of the dependency relations in the language. The languages are sorted on the precedence probabilities from low to high. RS: RSUnpert, TG: TGUnpert, PTS: PTSPertRS

ratio of the number of times the dependents with that relation appear before their heads and the total number of times the relation occurs in the data.

In our experiments, the precedence probabilities

Dep Rel	RSU-npert	TGU-npert	PTSPert-RS (P=0.2)
case	69.2	72.0	**75.6**
nmod	26.2	27.2	**29.0**
nsubj	52.5	51.2	**56.6**
amod	74.1	78.7	**79.4**
obl	40.8	38.8	**47.3**
advmod	63.2	60.7	**63.7**
obj	46.0	46.1	**48.8**
aux	59.5	72.4	**78.9**
mark	62.3	61.7	**63.5**
cc	71.3	71.1	**71.9**
acl	23.6	21.5	**24.8**
advcl	**32.5**	29.5	32.2
det	79.3	**86.2**	82.5
cop	57.4	**61.6**	60.1
nummod	65.3	**68.1**	67.7
compound	**36.5**	34.7	33.2
xcomp	34.9	**39.5**	34.9
flat	34.9	**35.6**	35.1

Table 3: Dependency-wise average accuracies of RSUnpert, TGUnpert and PTSPertRS (P=0.2).

of the relations in the source and target languages are estimated from the corresponding training and test sets respectively. Note that we have used these estimates for analysis of the results only.

In Figure 3 we compare the gain in LAS of PTSPertRS over RSUnpert parser corresponding to 4 different dependency relations over all the target languages. The dependency relations are chosen such that two are short distance relations (intra-phrase): *case* and *auxiliary* and two are relatively long-distance relations (inter-phrase): *nsubj* and *obl*.

For all the four dependency relations, we observe that the gains in performance of PTSPertRS over TGUnpert increases with the increase in the difference of precedence probability of the relations in the languages from that of English.

We also observe significant improvement in the performance of the PTSPertRS parsers over TGUnpert in case of the *nsubj* and *obl* for most of the language. Only in case of *fi* and *ko* languages, both RSUnpert and TGUnpert perform better than PTSPertRS in terms of the *nsubj* relation.

It is also observed that PTSPertRS performs significantly better than RSUnpert and TGUnpert in terms of the *aux* and *case* relations for the languages in which the precedence probabilities of the relations are different from that of English.

5.2.2 PTSPertRS with Variable Perturbation Probability Values

The results on PTSPertRS discussed above correspond to a single perturbation probability value

applied on all the dependency relations. However, we observed that the best accuracies corresponding to different dependency relations were achieved at different P values. Thus we hypothesize that perturbing the dependents of different dependency relations by different amounts might be more helpful. We try to get an estimate of the perturbation probabilities corresponding different dependency relations from the performance of the PTSPertRS models trained using augmented treebanks perturbed with different perturbation probability values on the test set of a small number of languages. For this, we selected a random subset of 9 languages from the 29 languages. The 9 reference languages are *es, sl, he, id, sv, de, et, ar* and *hi*.

The steps for obtaining the probability value corresponding to a dependency relation are as follows;

- Corresponding to each P value in $\{0.0, 0.1, 0.2, 0.3, 0, 4, 0.5\}$, we find the average accuracy for the dependency relation over the 9 languages.

- We take the P value for the dependency relation for which the highest average accuracy is observed.

In Table 4 we present the perturbation probability values used for the different dependency relations. We apply these perturbation prob-

Dep Rels	Pert Prob
appos, parataxis, goeswith,flat, discourse, list, iobj,expl	0.0
det, cop, acl, advcl, mark, cc, compound	0.1
aux, ccomp, amod, obl, nummod, advmod, acl	0.2
nsubj, nmod, discourse, vocative	0.3
obj	0.4
case, csubj	0.5

Table 4: Perturbation probability values corresponding to the different dependency relations.

ability values corresponding to the different dependency relations to perturb the source language parse trees.

In Table 5 we present the performances of TGUnpert, RSUnpert, PTSPertRS with fixed P values and the PTSPertRS with variable P values averaged over all the 9 reference languages, 29 target languages and the 20 held-out languages respectively. On the set of the held-out 20 target

No. of languages		RSU-npert	TGU-npert	PTS-Pert (0.2)	PTS-Pert (Var. P)
9 reference languages	UAS	57.9	60.1	63.3	**64.7**
	LAS	47.7	50.0	53.2	**55.1**
20 held-out languages	UAS	64.0	65.5	67.6	**68.6**
	LAS	53.9	55.5	57.4	**58.6**
29 target languages	UAS	62.1	62.7	66.3	**67.4**
	LAS	52.0	52.5	56.1	**57.5**

Table 5: Average %UAS/%LAS over different sets of target languages for RSUnpert, TGUnpert, PTSPertRS (P=0.2) and PTSPertRS with variable P.

languages, we observe an improvement of 1.6% UAS and 2.16% LAS over the best single perturbation probability value of ($P = 0.2$) on the 20 languages. On the set of all the 29 languages also, this perturbation approach results in an overall improvement of 1.66% UAS and 2.49% LAS over the best single perturbation value ($P = 0.2$).

5.3 Results with Hindi as Source Language

We report here a summary of the results for Hindi as the source language. The variable perturbation probability values were derived from the following languages: *es, sl, he, id, sv, de, et, ar* and *en*.

In Table 6 we present the results corresponding to the different transfer approaches averaged over 29 target languages. We observe that PTSPertRS with different values of P outperform RSUnpert and TGUnpert. The best PTSPertRS result is achieved at P=0.3. PTSPertRS with variable P values also performs better than fixed P values. We observe that PTSPert with P=0.3 and variable P performs better than RSUnpert and TGUnpert for 27 out of 29 languages except *ko* and *ja*. We observe that *ko* and *ja* are syntactically quite close to Hindi and hence a parser model trained on unperturbed treebanks perform better than their perturbed versions.

In Table 7 we compare the performance of RSUnpert, TGUnpert, PTSPertRS with the $P = 0.3$ and PTSPertRS with variable P value averaged over all the 9 reference languages, 29 target languages and the 20 held-out languages respectively. We observe that PTSPertRS with variable P values gives the best results.

In Table 8 we report the average performance of RSUnpert, TGUnpert, PTSPertRS with the $P = 0.3$ and PTSPertRS with variable P values on Tamil (ta), Telugu (te), Urdu (ur) and Marathi (mr) languages for which treebanks are available in UD

v2.2.

We observe that on an average over the four Indian languages, the best UAS and LAS scores are achieved for RSUnpert and TGUnpert respectively. Since the distribution of the dependents with respect to their heads for different dependency relations in the Indian languages are similar to that of Hindi, the best results are obtained for the parsers trained using unperturbed source treebank. This observation is in coherence with the results in English and Hindi where RSUnpert trained on unperturbed treebanks yield better results than PTSPert for the languages syntactically similar to the corresponding sources languages i.e. *no* and *sv* for English and *ko* and *ja* for Hindi.

5.4 Performance over Other Source Languages

We show that our perturbation approach enhances the performance of the "order-free" model proposed by Ahmad et al. (2019) for most of the 29 source languages. For this, we trained the parser model with transformer-based encoder and graph-based decoder using both unperturbed source language treebanks and PTSPert treebanks perturbed using P=0.1. The performance of the model trained using unperturbed treebank is taken as baseline. Following Ahmad et al. (2019), we trained the models using the first 4000 parse trees of each of the source language treebanks.
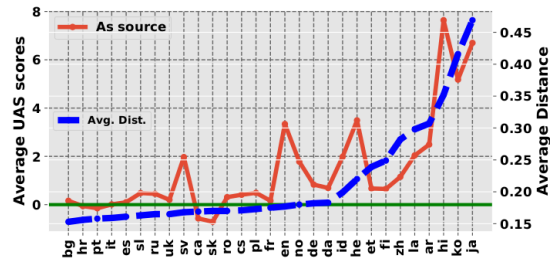

Figure 4: The red curve indicates average improvement over baseline for a language as source. The blue curve and the right y-axis indicates the average distance of a language from the rest.

In Figure 4, for each language as a source, we show the average improvement over all the target languages in cross-lingual performance of the parser trained using perturbed treebank. The languages are sorted according to their average syntactic distance from the other languages.

We observe that perturbation improves the average performance of the transfer parsers for the

	RSUnpert	TGUnpert	PTSPertRS					
			0.1	0.2	0.3	0.4	0.5	Var P
UAS	36.2	40.6	49.9	52.3	52.9	52.7	52.7	**54.0**
LAS	26.1	30.2	37.7	39.6	39.9	39.4	39.7	**41.2**

Table 6: Average UAS%/LAS% of different transfer parser approaches with Hindi as the source language.

No. of langs		RSU-npert	TGU-npert	PTS-Pert (0.3)	PTS-Pert (Var. P)
9 reference languages	UAS	31.8	36.7	50.8	**52.4**
	LAS	22.4	27.1	37.7	**39.6**
20 held-out languages	UAS	38.2	42.3	53.8	**54.7**
	LAS	27.8	31.6	40.8	**42.0**
29 target languages	UAS	36.2	40.5	52.9	**54.0**
	LAS	26.1	30.2	39.9	**41.2**

Table 7: Average %UAS/%LAS over different sets of target languages for different parsing approaches with Hindi as source language.

	RSU-npert	TGU-npert	PTS-Pert (0.3)	PTS-Pert (Var. P)
UAS	**75.9**	74.9	73.9	74.4
LAS	55.6	**55.9**	54.8	55.5

Table 8: Average %UAS/%LAS over *ta, te, mr* AND *ur* for different parsing approaches with Hindi as source language.

source languages except *pt*, *sk* and *ca*. The Pearson correlation coefficient of the average improvements with the languages as source with respect to the average distance from other languages is 0.82 indicating that the improvement due to perturbation is strongly correlated with the average distance from the target languages.

6 Conclusion

In this paper propose an approach for introducing perturbation in the source language treebank to improve single source target language independent cross-lingual transfer parsing. We show that this approach indeed helps to improve the performance of the transferred parsers over models trained using only source language treebanks.

References

Wasi Uddin Ahmad, Zhisong Zhang, Zuezhe 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 the 2019 Conference of the NAACL: HLT*.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *International Conference on Learning Representations*.

Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. 2016. Zero-resource dependency parsing: Boosting delexicalized cross-lingual transfer with linguistic knowledge. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 119–130.

Riyaz A. Bhat, Irshad Bhat, and Dipti Sharma. 2017. Leveraging newswire treebanks for parsing conversational data with argument scrambling. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 61–66, Pisa, Italy. Association for Computational Linguistics.

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.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087.

Ayan Das and Sudeshna Sarkar. 2019. Transform, combine, and transfer: Delexicalized transfer parser for low-resource languages. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 19(1):4:1–4:30.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

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 ACL and the 7th IJCNLP (Volume 1: Long Papers)*, pages 1234–1244, Beijing, China. Association for Computational Linguistics.

Martin Haspelmath. 2005. *The world atlas of language structures / edited by Martin Haspelmath ... [et al.]*. Oxford University Press Oxford.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*.

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

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.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the ACL (Volume 2: Short Papers)*, volume 2, pages 92–97.

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

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the ACL: Long Papers - Volume 1*, ACL '12, pages 629–637, Stroudsburg, PA, USA. Association for Computational Linguistics.

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. *CoRR*, abs/1903.05683.

Rudolf Rosa and Zdenek Zabokrtsky. 2015. Klcpos3-a language similarity measure for delexicalized parser transfer. In *Proceedings of the 53rd Annual Meeting of the ACL and the 7th IJCNLP (Volume 2: Short Papers)*, volume 2, pages 243–249.

M. Schuster and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. *arXiv preprint arXiv:1902.09492*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *CoRR*, abs/1803.02155.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Anders Søgaard. 2011. Data point selection for cross-language adaptation of dependency parsers. In *Proceedings of the 49th Annual Meeting of the ACL: HLT: short papers-Volume 2*, pages 682–686. Association for Computational Linguistics.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. Proceedings of the 2013 Conference of the NAACL: HLT.

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 NAACL: HLT*, pages 477–487. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

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. 2018. Synthetic data made to order: The case of parsing. In *Proceedings of the 2018 Conference on EMNLP*, pages 1325–1337.

Min Xiao and Yuhong Guo. 2014. Distributed word representation learning for cross-lingual dependency parsing. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 119–129.

D. Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. *NLP for Less Privileged Languages*, pages 35 – 35.

Yuan Zhang and Regina Barzilay. 2015. Hierarchical low-rank tensors for multilingual transfer parsing. Association for Computational Linguistics.

A Appendices

A.1 Language Name Abbreviations

en - English, no - Norwegian, sv - Swedish, fr - French, pt - Portuguese, da - Danish, es - Spanish, it - Italian, hr - Croatian, ca - Catalan, pl - Polish, uk - Ukranian, sl - Slovenian, bg - Bulgarian, ru - Russian, de - German, he - Hebrew, cs - Czech, ro - Romanian, sk - Slovak, id - Indonesian, fi - Finnish, et - Estonian, zh - Chinese, ar - Arabic, la - Latin, ko - Korean, hi - Hindi, ja - Japanese.