Cross-Lingual Dependency Parsing Using Code-Mixed TreeBank

Meishan Zhang[1]   and   **Yue Zhang**[2*] and   **Guohong Fu**[3]

1. School of New Media and Communication, Tianjin University, China
2. Institute of Advanced Technology, Westlake Institute for Advanced Study
3. Institute of Artificial Intelligence, Soochow University, China

```
mason.zms@gmail.com,
zhangyue@westlake.edu.cn,
ghfu@hotmail.com
```

Abstract

Treebank translation is a promising method for cross-lingual transfer of syntactic dependency knowledge. The basic idea is to map dependency arcs from a source treebank to its target translation according to word alignments. This method, however, can suffer from imperfect alignment between source and target words. To address this problem, we investigate syntactic transfer by code mixing, translating only confident words in a source treebank. Cross-lingual word embeddings are leveraged for transferring syntactic knowledge to the target from the resulting code-mixed treebank. Experiments on University Dependency Treebanks show that code-mixed treebanks are more effective than translated treebanks, giving highly competitive performances among cross-lingual parsing methods.

1 Introduction

Treebank translation (Tiedemann et al., 2014; Tiedemann, 2015; Tiedemann and Agić, 2016) has been considered as a method for cross-lingual syntactic transfer. Take dependency grammar for instance. Given a source treebank, machine translation is used to find target translations of its sentences. Then word alignment is used to find mappings between source and target words, so that source syntactic dependencies can be projected to the target translations. Following, a post-processing step is applied by removing unaligned target words, in order to ensure that the resulting target syntax forms a valid dependency tree, The method has shown promising performance for unsupervised cross-lingual dependency parsing among transfer methods (McDonald et al., 2011; Täckström et al., 2012; Rasooli and Collins, 2015; Guo et al., 2016b).

(a) full-scale translation.

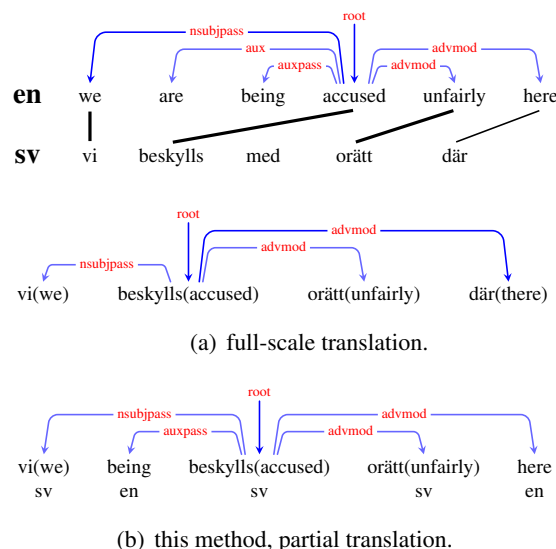(b) this method, partial translation.

Figure 1: An example to illustrate our method, where the source and target languages are English (en) and Swedish (sv), respectively.

The treebank translation method, however, suffers from various sources of noise. For example, machine translation errors directly affect the resulting treebank, by introducing ungrammatical word sequences. In addition, the alignments between source and target words may not be isomorphic due to inherent differences between languages or paraphrasing during translation. For example, in the case of Figure 1, the English words "are" and "being", and the Swedish word "med", do not have corresponding word-level translation. In addition, it can be perfect to express "as soon as they can" using "very quickly" in a translation, which looses word alignment information because of the longer span. Finally, errors in automatic word alignments can also bring noise. Such alignment errors can directly affect grammaticality of the resulting target treebank due to deletion of unaligned words during post-processing, or cause

---

*Corresponding author.

lost or mistaken dependency arcs.

We consider a different approach for translation-based syntactic knowledge transfer, which aims at making the best use of source syntax with the minimum noise being introduced. To this end, we leverage recent advances in cross-lingual word representations, such as cross-lingual word clusters (Täckström et al., 2012) and cross-lingual word embeddings (Guo et al., 2015), which allow words from different languages to reside within a consistent feature vector space according to structural similarities between words. Thus, they offer a bridge on the lexical level between different languages (Ammar et al., 2016).

A cross-lingual model can be trained by directly using cross-lingual word representations on a source treebank (Guo et al., 2015). Using this method, knowledge transfer can be achieved on the level of token correspondences. We take this approach as a naive baseline. To further introduce structural feature transfer, we transform a source treebank into a code-mixed treebank by considering word alignments between a source sentence and its machine translation. In particular, source words with highly confident target alignments are translated into target words by consulting the machine translation output, so that target word vectors can be directly used for learning target syntax. In addition, continuous spans of target words of the code-mixed treebank are reordered according to the translation, so that target grammaticality can be exploited to the maximum potent.

We conduct experiments on Universal Dependency Treebanks (v2.0) (McDonald et al., 2013; Nivre et al., 2016). The results show that a code-mixed treebank can bring significantly better performance compared to a fully translated treebank, resulting in averaged improvements of 4.30 points on LAS. The code and related data will be released publicly available under Apache License 2.0.[1]

2   Related Work

Existing work on cross-lingual transfer can be classified into two categories. The first aims to train a dependency parsing model on source treebanks (McDonald et al., 2011; Guo et al., 2016a,b), or their adapted versions (Zhao et al., 2009; Tiedemann et al., 2014; Wang et al., 2017) in the target language. The second category,

namely annotation projection, aims to produce a set of large-scale training instances full of automatic dependencies by parsing parallel sentences (Hwa et al., 2005; Rasooli and Collins, 2015). The two broad methods are orthogonal to each other, and can both make use of the lexicalized dependency models trained with cross-lingual word representations (Rasooli and Collins, 2017, 2019).

Source Treebank Adaption. There has been much work on unsupervised cross-lingual dependency parsing by direct source treebank transferring. Several researchers investigate delexicalized models where only non-lexical features are used in the models (Zeman and Resnik, 2008; Cohen et al., 2011; McDonald et al., 2011; Naseem et al., 2012; Täckström et al., 2013; Rosa and Zabokrtsky, 2015). All the features in these models are language independent, and are consistent across languages and treebanks. Thus they can be applied into target languages directly.

Subsequent research proposes to exploit lexicalized features to enhance the parsing models, by resorting to cross-lingual word representations (Täckström et al., 2012; Guo et al., 2015; Duong et al., 2015b,a; Zhang and Barzilay, 2015; Guo et al., 2016b; Ammar et al., 2016; Wick et al., 2016; de Lhoneux et al., 2018). Cross-lingual word clusters and cross-lingual word embeddings are two main sources of features for transferring knowledge between source and target language sentences. These studies enable us to train lexicalized models on code-mixed treebanks as well. Thus here we integrate the cross-lingual word representations as well, which gives more direct interaction between source and target words.

Our work follows another mainstream method of this line of work, namely treebank translation (Tiedemann et al., 2014; Tiedemann, 2015; Tiedemann and Agić, 2016), which aims to adapt an annotated source treebank into the target language by machine translation. In addition, the target-side sentences are produced by machine translation. Previous work aims to build a well-formed tree (Tiedemann and Agić, 2016) from source dependencies, solving word alignment conflicts by heuristic rules. In contrast, we use partial translation instead to avoid unnecessary noise.

Annotation Projection. The annotation projection approach relies on a set of parallel sentences between the source and target languages (Hwa et al., 2005; Ganchev et al., 2009). In par-
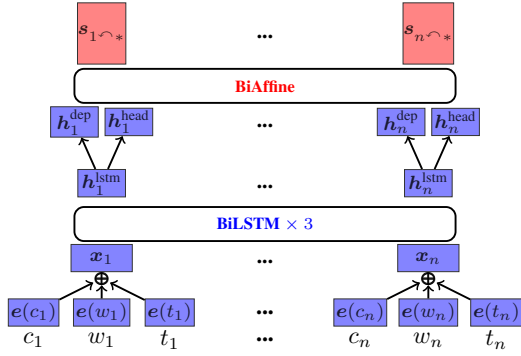
---

[1] https://github.com/zhangmeishan/CodeMixedTreebank

Figure 2: The overall architecture of the BiAffine parser.

ticular, a source parser trained on the source treebank is used to parse the source-side sentences of the parallel corpus. The source dependencies are then projected onto the target sentences according to word alignments. Different strategies can be applied for the dependency projection task (Ma and Xia, 2014; Rasooli and Collins, 2015; Xiao and Guo, 2015; Agić et al., 2016; Schlichtkrull and Søgaard, 2017). For example, one can project only dependency arcs whose words are aligned to target-side words with high confidence (Lacroix et al., 2016). The resulting treebank can be highly noisy due to the auto-parsed source dependency trees. Recently Lacroix et al. (2016) and Rasooli and Collins (2017) propose to filter the results from the large-scale parallel corpus. Our work is different in that the source dependencies are from gold-standard treebanks.

3 Dependency Parsing

We adopt a state-of-the-art neural BiAffine parser (Dozat and Manning, 2016) as the baseline, which has achieved competitive performances for dependency parsing. The overall architecture is shown in Figure 2. Given an input sentence $w_1 \cdots w_n$, the model finds its embedding representations of each word $x_1 \cdots x_n$, where

$$x_i = e(w_i) \oplus e(c_i) \oplus e(t_i) \quad (1)$$

Here c_i denotes the word cluster of w_i, and t_i denotes the POS tag. We exploit cross-lingual word embeddings, clusters and also universal POS tags, respectively, which are consistent across language and treebanks.

A three-layer deep bidirectional long short term memory (LSTM) neural structure is applied on $x_1 \cdots x_n$ to obtain hidden vectors

$h_1^{\text{LSTM}} \cdots h_n^{\text{LSTM}}$. For head finding, two non-linear feed-forward neural layers are used on $h_1^{\text{LSTM}} \cdots h_n^{\text{LSTM}}$ to obtain $h_1^{\text{dep}} \cdots h_n^{\text{dep}}$ and $h_1^{\text{head}} \cdots h_n^{\text{head}}$. We compute the score for each dependency $i \frown j$ by:

$$s_{i \frown j} = \text{BiAffine}(h_i^{\text{dep}}, h_j^{\text{head}}) \quad (2)$$

The above process is also used for scoring a labeled dependency $i \overset{1}{\frown} j$, by extending the 1-dim vector s into L dims, where L is the total number of dependency labels.

When all scores are ready, a softmax function is used at each position for all candidate heads and labels, and then the normalized scores are used for decoding and training. For decoding, we exploit the MST algorithm to ensure tree-structural outputs. For training, we accumulate the cross-entropy loss at the word-level by treating the normalized scores as prediction probabilities. The reader is referred to Dozat and Manning (2016) for more details.

4 Code-Mixed Treebank Translation

We derive code-mixed trees from source dependency trees by partial translation, projecting words and the corresponding dependencies having high-confidence alignments with machine-translated target sentences. Our approach assumes that sentence level translations and alignment probabilities are available. The motivation is to reduce noise induced by problematic word alignments.

We adopt the word-level alignment strategy, which has been demonstrated as effective as phrase-level alignment yet much simpler (Tiedemann et al., 2014; Tiedemann, 2015; Tiedemann and Agić, 2016). Give a source sentence $e_1 \cdots e_n$ and its target language translation $f_1 \cdots f_m$, $p(e_i | f_j)$ denotes the probability of word f_j being aligned with e_i ($0 \leq i \leq n$ and $0 < j \leq m$), where e_0 denotes a null word, indicating the no alignment probability for one target word.

The translation process can be conducted by three steps:

(1) word substitution, which incrementally substitutes the source words with the target translations;

(2) word deletion, which removes several unaligned source words;

Algorithm 1 The process of tree translation.

Input: $e_1 \cdots e_n$, $f_1 \cdots f_m$, T_e, $\{p(e|f)\}$, λ
Output: T_{mix}

1: Let $T_{mix} = T_e$.
2: **for** $j \in [1, m]$ **do**
3:     $a_j, p_j \leftarrow \operatorname{argmax}_i p(e_i|f_j), \max_i p(e_i|f_j)$
4: **end for**
5: $S_{sub} \leftarrow \text{SELECT}(\{\langle f_j, a_j, \boldsymbol{v}=p_j\rangle\}, \lambda)$
6: **for** $f_j, a_j \in S_{sub}$ **do**
7:     $T_{mix} \leftarrow \text{SUBSTITUTE}(f_j, a_j, T_{mix})$
8: **end for**
9: $D = [1, n] - \{a_j\}|_{j \in [1,m]}$
10: **for** $i \in D$ **do**
11:     $r_i = \sum_j p(e_i|f_j)$
12: **end for**
13: $S_{del} \leftarrow \text{SELECT}(\{\langle e_i, \boldsymbol{v}=-r_i\rangle\}, \lambda)$
14: **for** $e_i \in S_{del}$ **do**
15:     $T_{mix} \leftarrow \text{DELETE}(e_i, T_{mix})$
16: **end for**
17: $T_{mix} \leftarrow \text{REORDER}(T_{mix}, f_1 \cdots f_m)$

(3) sentence reordering, which reorders the partially translated sentence, ensuring local target language word order.

Algorithm 1 shows pseudocode for code-mixing tree translation, where line 1-8 denotes the first step, line 9-16 denotes the second and line 17 denotes the last step.

There is a hyper-parameter λ to control the overall ratio of translation. The function SELECT(S, λ) is to obtain a subset of S by ratio λ with top element values as indicated by $\boldsymbol{v}$ inside S. If $\lambda = 0$, it is still the source language dependence tree, since no source word is substituted or deleted. In this condition, our method is equal to Guo et al. (2015) by bridging source and target dependency parsing with universal word representations. If $\lambda = 1$, the resulting tree is a fully translated target dependency tree, as all words are target language produced by translation. In this setting, our method is equal to Tiedemann and Agić (2016) where the only difference the our baseline parsing model. Thus our method can be regarded as a generalization of both source-side training (Guo et al., 2015) and fully translated target training (Tiedemann and Agić, 2016) with fine-grained control over translation confidence.
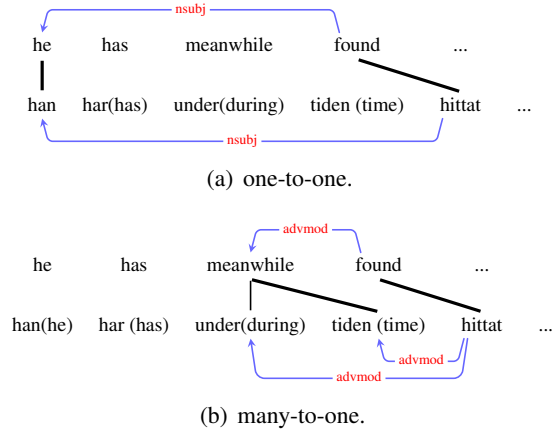

(a) one-to-one.

(b) many-to-one.

Figure 3: Examples of word substitution, where the thicker lines indicates more confident alignments.

4.1 Word Substitution

Word substitution is the key step for producing a target treebank. We first obtain the most confidently aligned source word e_{a_j} for each target word f_j as well as their alignment probability $p_j = p(e_{a_j}|f_j)$, as shown by line 3 in Algorithm 1. Then we sort the target words by these alignment probabilities, choosing the top $\lceil m\lambda \rceil$ words with highest alignment probabilities for substitution. The sorting and choosing is reflected in line line 5 of Algorithm 1. Finally for each chosen word f_j and its aligned source word e_{a_j}, we replace the source word e_{a_j} by f_j, as shown by line 7 in Algorithm 1.

One key of the substitution is to maintain the corresponding dependency structures. If e_{a_j} and f_j bares a one-one mapping, with no other target word being aligned with e_{a_j}, the source dependencies are kept unchanged, as shown by Figure 3(a). While if two or more words (i.e., $f_{j_1}, \cdots, f_{j_k}(j_1 < \cdots < j_k)$) are aligned to e_{a_j}, we simply link all words to e_{a_j}, with the same dependency label as the original dependency arc. Figure 3(b) illustrates this condition. Both the Swedish words "under" and "tiden" are headed to "hittat" (the Swedish translation of English word "found") by the dependency label "advmod" inherited from the source English side . Note that the POS tags of the substituted words are the same as the corresponding source words.

4.2 Word Deletion

There can be source words to which no target word is aligned. These words are typically functional words belonging to source language only, such as
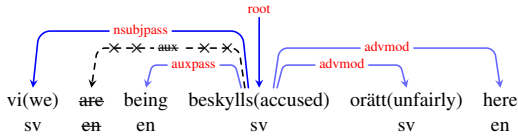
Figure 4: An example of word deletion.

(a) full sentence

(b) two spans

Figure 5: Two examples of sentence reorder.

"the", "are" and "have". We remove such words to produce dependency trees that are close in syntax to the target language.

In particular, we accumulate the probabilities of $p(e_i|f_j)$ for the source word e_i who has no aligned target word:

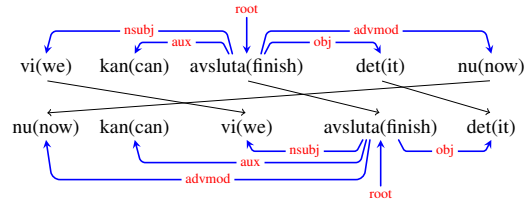$$r_i = \sum_j p(e_i|f_j) \quad \text{(line 11 in Algorithm 1),}$$

where we traverse all target words to sum their alignment probabilities with e_i. The value of r_i can be interpreted as the confidence score of retention. The words with lower retention scores should be more preferred to be deleted, as these words have lower probabilities aligning with some word of the target language sentence. Concretely, we collect all source words with no aligned target words, computing their retention values, and the selecting a subset of these words with the lowest retention values by the hyper-parameter λ (line 13 in Algorithm 1). Finally we delete all the selected words (line 15 in Algorithm 1).

Figure 4 shows an example of word deletion. The two words "are" and "being" both have no aligned words in the other side, and meanwhile "are" has a lower retention score compared with "being".[2] Thus the source word "are" is prefer to be deleted. In most cases, the deleted words are leaf nodes, which can be unattached to the resulted dependency tree and deleted them directly. In case of exceptions, we simply reset the corresponding heads of its child nodes by the head of e_i (i.e., h_{e_i}) instead. For example, a dependency $w_i^\frown e_i$ is changed into $w_i^\frown h_{e_i}$.
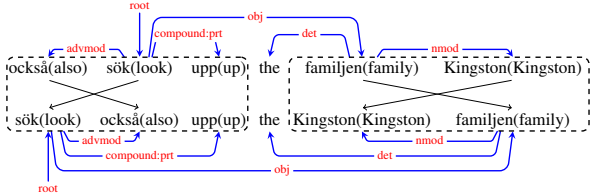
4.3 Sentence Reordering

Continuous target spans are reordered to make the final code-mixed sentence contain grammatical phrases in the target language. Figure 5(a) shows one example of full sentence reordering. We can see that the word order by word-level substitutions on the source words is different with the

---

[2]Both the two words are only related to the Swedish word "med", but $p(\text{are}|\text{med})$ is slightly lower.

order of the machine-translated sentence. Thus we adjust the leaf nodes, letting the word order strictly follow the machine-translated sentence order For example, the word "vi" is moved from the first position into the third position, and similarly the word "mu" is moved from the last position into the first position.

Concretely, we perform the word reorder by the span level, extracting all the continuous spans of target words, because the target language words may be interrupted by source language words. Then we reorder the words in each span according to their order in the machine translation outputs. Figure 5(b) shows another example, where there are two spans separated by the English word "the". Each span is reordered individually. We do not consider the inconsistent orders inter the spans in this work. Note that this step does not change any dependency arc between words.

5 Experiments

We conduct experiments to verify the effectiveness of our proposed models in this section.

5.1 Settings

Our experiments are conducted on the Google Universal Dependency Treebanks (v2.0) (McDonald et al., 2013; Nivre et al., 2016), using English as the source language, and choosing six languages, including Spanish (ES), German (DE), French (FR), Italian (IT), Portuguese (PT) and Swedish (sv), as the target languages. Google

Translate[3] is used to translate the sentences in the English training set into other languages. In order to generate high-quality word-level alignments, we merge the translated sentence pairs and the parallel data of EuroParl (Koehn, 2005) to obtain word alignments. We use the fastAlign tool (Dyer et al., 2013) to obtain word alignments.

We use the cross-lingual word embeddings and clusters by Guo et al. (2016b) for the baseline system. The dimension size of word embeddings is 50 and the word cluster number across of all languages is 256.

For network building and training, we use the same setting as Dozat and Manning (2016), including the dimensional sizes, the dropout ratio, as well as the parameter optimization method. We assume that no labeled corpus is available for the target language. Thus training is performed for 50 iterations over the whole training data without early-stopping.

To evaluate dependency parsing performances, we adopt UAS and LAS as the major metrics, which indicate the accuracies of unlabeled dependencies and labeled dependencies, respectively. We ignore the punctuation words during evaluation following previous work. We run each experiment 10 times and report the averaged results.

5.2   Models

We compare performances on the following models:

- Delex (McDonald et al., 2013): The delexicalized BiAffine model without cross-lingual word embeddings and clusters.

- Src (Guo et al., 2015): The BiAffine model trained on the source English treebank only.

- PartProj (Lacroix et al., 2016): The Bi-Affine model trained on the corpus by projecting only the source dependencies involving high-confidence alignments into target sentences. Note that the baseline only draws the idea from Lacroix et al. (2016), and the two models are significant different in fact.

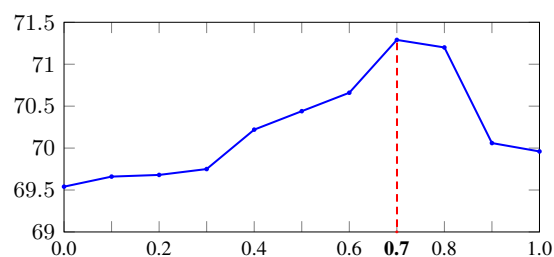- Tgt (Tiedemann and Agić, 2016): The Bi-Affine model trained on the fully translated target treebank only.

---

[3]https://translate.google.com/ at Oct, 2018

Figure 6: The LAS relative to the translation ratio λ.

Model	UAS	LAS
Src	79.52	69.54
Tgt	79.34	69.96
Mix	**80.33**	**71.29**
Src + Tgt	80.12	71.16
Src + Mix	**80.91**	**71.73**

Table 1: Experiments of corpus mixing.

- Src+Tgt: The BiAffine model trained on the combination dataset of the source and fully translated target treebanks.

- Mix: The BiAffine model trained on the code-mixed treebank only.

- Src+Mix: The BiAffine model trained on the combination dataset of the source and code-mixed treebanks.

The Src and Tgt methods have been discussed in Section 4. The PartProj model is another way to leverage imperfect word alignments (Lacroix et al., 2016). The training corpus of PartProj may be incomplete dependency trees with a number of words missing heads, because no word is deleted from machine translation outputs. The POS tags of words in PartProj with low-confidence alignments are obtained by a supervised POS tagger (Yang et al., 2018) trained on the corresponding universal treebank.

5.3   Development Results

We conduct several developmental experiments on the Swedish dataset to examine important factors to our model.

5.3.1   Influence of The Translation Ratio λ

Our model has an important hyper-parameter λ to control the percentage of translation. Figure 6 shows the influence of this factor, where the percentages increase from 0 to 1 by intervals of 0.1. A

1002

Model	UAS	LAS
Mix	**80.33**	**71.29**
−Sentence Reordering	79.79	70.47
−Word Deletion	79.82	70.64
−Both	79.46	69.59

Table 2: Ablation experiments.

λ of 0 gives our baseline by using the source treebank only. As the λ grows, more source words are translated into the target. We can see that the performance improves after translating some source dependencies into the target, demonstrating the effectiveness of syntactic transferring. The performance reaches the peak when $\lambda = 0.7$, but there is a significant drop when λ grows from 0.8 to 0.9. This can be because the newly added dependency arc projections are mostly noisy. This sharp decrease indicates that noise from low-confidence word alignments can have strong impact on the performance. According to the results, we adopt $\lambda = 0.7$ for code-mixed treebanking.

5.3.2 Mixing with Source TreeBank

We investigate the effectiveness of the source treebank by merging it into the translated treebanks. First, we show the model performances of Src, Tgt and Mix, which are trained on the individual treebanks, respectively. Then we merge the source treebank with the two translated treebanks, and show the results trained on the merging corpora. Table 1 shows the results. According to the results, we can find that the source treebank is complementary with the translated treebanks. Noticeably, although Src + Mix gives the best performance, its improvement over Mix is relatively smaller than that of Src + Tgt over Tgt. This is reasonable as the code-mixed treebank contains relatively more source treebank content than the fully translated target treebank.

5.3.3 Ablation Studies

The overall translation is conducted by three steps as mentioned in Section 4, where the first word substitution is compulsory, and the remaining two steps aim to build better mixed dependency trees. Here we conduct ablation studies to test the effectiveness of word deletion and sentence reordering. Table 2 shows the experimental results. We can see both steps are important for dependency tree translation. Without word deletion and sentence reordering, the mix model shows decreases of 0.82 and 0.65 on LAS, respectively. If both are removed, the performance is only comparable with the baseline src model (see Table 1).

5.4 Final Results

We show the final results of our proposed models in Table 3. As shown, the model Tgt gives better averaged performance compared to Src. However, its results on French and Italian are slightly worse, which indicates that noise from translation impacts the quality of the projected treebank. The model Mix gives much better performance compared with Src, demonstrating the effectiveness of structural transfer. Mix also outperforms Tgt significantly, obtaining average increases of $80.93 - 76.37 = 4.56$ points on UAS and $72.87 - 68.28 = 4.59$ points on LAS, respectively. The best setting is the model Src + Mix, trained on the combined corpus of the source and code-mixed treebanks, which gives better performance than solely code-mixed treebanks.

By comparing with the delexicalized model Delex, we can see that lexicalized features are highly useful for cross-lingual transfer. For the PartProj model, we conduct preliminary experiments on Swedish to tune the ratio of the projected dependencies. The results show that the difference is very small ($\delta = 0.24$ for UAS) between 0.9 to 1.0, and the performance degrades significantly as the ratio decreases below 0.9. The observation indicates that this method is probably not effective for filtering low-confidence word alignments. The final results confirm our hypothesis. As shown in Table 3, the PartProj model gives only comparable performance with Src. One possible reason may be the unremoved target words (if the words are removed, the PartProj model with ratio 1.0 will be identical to Tgt), which have been demonstrated noisy previously (Tiedemann and Agić, 2016).

5.5 Comparison with Previous Work

We compare our method with previous work in the literature. Table 4 shows the results, where the UAS values are reported. Our model denoted by **This** refers to the model of Src + Mix. Note that these models are not directly comparable due to the setting and baseline parser differences. The first block shows several models by directly transferring gold-standard source treebank knowledge into the target side, including the models of

Lang.	Delex		PartProj		Src		Tgt		Src + Tgt		Mix		**Src + Mix**	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
DE	64.10	53.77	69.90	61.28	66.87	57.46	70.84	62.30	72.41	63.74	71.41	63.46	**72.78**	**64.38**
ES	71.53	63.33	75.81	66.83	75.63	65.85	76.49	67.39	77.00	67.95	81.18	71.80	**81.44**	**71.66**
FR	75.13	67.26	75.54	67.63	78.13	70.63	76.91	69.39	78.75	71.17	83.20	76.32	**83.77**	**76.48**
IT	77.71	69.27	77.71	69.27	81.11	72.83	79.30	71.65	81.56	74.09	85.30	77.43	**86.13**	**78.38**
PT	74.03	67.70	79.44	71.30	77.37	69.36	78.32	70.67	79.73	71.84	83.54	75.34	**84.05**	**75.89**
AVG	72.50	64.27	75.68	67.26	75.82	67.23	76.37	68.28	77.89	69.76	80.93	72.87	**81.63**	**73.36**

Table 3: Final results.

Model	DE	ES	FR	IT	PT
TreeBank Transferring					
This	72.78	**81.44**	**83.77**	**86.13**	**84.05**
Guo15	60.35	71.90	72.93	–	–
Guo16	65.01	79.00	77.69	78.49	81.86
TA16	**75.27**	76.85	79.21	–	–
Annotation Projection					
MX14	74.30	75.53	70.14	77.74	76.65
RC15	**79.68**	**80.86**	**82.72**	**83.67**	**82.07**
LA16	75.99	78.94	80.80	79.39	–
TreeBank Transferring + Annotation Projection					
RC17	**82.1**	**82.6**	**83.9**	**84.4**	**84.6**

Table 4: Comparison with previous work (UAS).

Guo15 (Guo et al., 2015), Guo16 (Guo et al., 2016b) and TA16 (Tiedemann and Agić, 2016). Our model gives the best performance with one exception on the German language. One possible reason may be that TA16 has exploited multiple sources of treebanks besides English.

The second block shows representative annotation projection models, including MX14 (Ma and Xia, 2014), RC15 (Rasooli and Collins, 2015), LA16. The models of annotation projection can be complementary with our work, since they build target training corpus from raw parallel texts. The best-performed results of the RC17 model (Rasooli and Collins, 2017) have demonstrated this point, which can be regarded as a combination of the dictionary-based treebank translation[4] (Zhao et al., 2009) and RC15.

5.6 Analysis

We conduct experimental analysis on the Spanish (ES) dataset to show the differences between the models of Src, Tgt and Mix.

---

[4]The method has been demonstrated worse than TA16 in Tiedemann and Agić (2016).
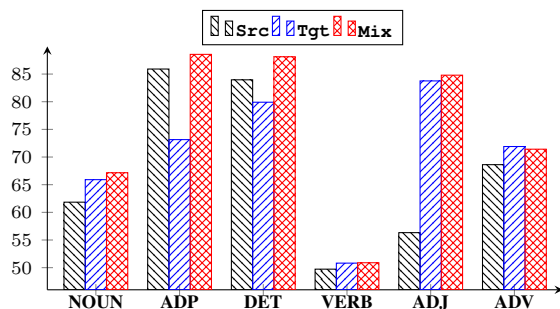

Figure 7: Performance relative to POS tags (F-score).

5.6.1 Performance Relative to POS Tags

Figure 7 show the F-scores of labeled dependencies on different POS tags. We list the six most representative POS tags. The Mix model achieves the best F-scores on 5 of 6 POS tags, with the only exception on tag ADV which has no significant difference with the Tgt model. The Mix and Tgt models are much better than the Src model as a whole, especially on the POS tag ADJ where an increase of over 20% has been achieved. In addition, we find that the Src model can significantly outperform the Tgt model on ADP and DET. For Spanish, ADP words are typically "de", "en", "con" and etc., which behave similarly to the English words such as "'s", "to" and "of". The Spanish words of DET include "el", "la", "su" and etc., which are similar to the English words such as "the" and "a". These words are highly ambiguous for automatic word alignment. The results indicate that our Mix model can better handle these word alignment noise, mitigating their negative influence of treebank translation, while the Tgt model suffers from such noise.

5.6.2 Performance Relative to Arc Distances

Figure 8 show the F-scores of labeled dependencies by different arc distances. Particularly we treat the root type as one special case. According
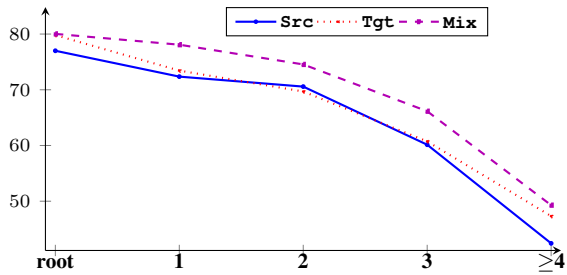
Figure 8: Performance relative to arc distances (F-score).

to the results, the `Mix` model performs the best over all distances, indicating its effectiveness on treebank transferring. The `Tgt` model achieves better performance than the `Src` model with one exception on distance 2. We look into the dependency patterns of distance 2 arcs further, finding that the dependency arc **ADP⌢*** accounts for over 30%, and it is the major source of errors. The finding is consistent with that on POS tags, denoting the effectiveness of the code-mixed treebank in handling noise. In addition, as the distance increases the performance drops gradually. The F-score of root dependency is the highest.

6 Conclusion

We proposed a new treebank translation method for unsupervised cross-lingual dependency parsing. Unlike previous work, which adopts full-scale translation for source dependency trees, we investigated partial translation instead, producing synthetic code-mixed treebanks. The method can better leverage imperfect word alignments between source and target sentence pairs, translating only high-confidence source sentential words, thus generating dependencies in high-quality. Experimental results on Universal Dependency Treebak v2.0 showed that partial translation is highly effective, and code-mixed treebanks can give significantly better results than full-scale translation.

Our method is complementary with several other methods for cross-lingual transfer, such as annotation projection, and thus can be further integrated with these methods.

Acknowledgments

References

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Martínez Alonso, Natalie Schluter, and Anders Søgaard. 2016. Multilingual projection for parsing truly low-resource languages. *TACL*, 4:301–312.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2016. Many languages, one parser. *TACL*, 4:431–444.

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.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015a. Cross-lingual transfer for unsupervised dependency parsing without parallel data. In *Proceedings of the CONLL*, pages 113–122.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015b. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd ACL*, volume 2, pages 845–850.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the NAACL*, pages 644–648.

Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. 2009. Dependency grammar induction via bitext projection constraints. In *Proceedings of the Joint Conference of ACL-IJCNLP*, pages 369–377.

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

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2016a. A distributed representation-based framework for cross-lingual transfer parsing. *Journal of Artificial Intelligence Research*, 55:995–1023.

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

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.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Ophélie Lacroix, Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. 2016. Frustratingly easy cross-lingual transfer for transition-based dependency parsing. In *Proceedings of the NAACL*, pages 1058–1063.

Miryam de Lhoneux, Johannes Bjerva, Isabelle Augenstein, and Anders Søgaard. 2018. Parameter sharing between dependency parsers for related languages. In *Proceedings of the EMNLP*, pages 4992–4997.

Xuezhe Ma and Fei Xia. 2014. Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization. In *Proceedings of ACL*, volume 1, pages 1337–1348.

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 ACL*, 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.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th ACL*, pages 629–637.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan T McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *LREC*.

Mohammad Sadegh Rasooli and Michael Collins. 2015. Density-driven cross-lingual transfer of dependency parsers. In *Proceedings of the 2015 EMNLP*, pages 328–338.

Mohammad Sadegh Rasooli and Michael Collins. 2017. Cross-lingual syntactic transfer with limited resources. *TACL*, 5:279–293.

Mohammad Sadegh Rasooli and Michael Collins. 2019. Low-resource syntactic transfer with unsupervised source reordering. *arXiv preprint arXiv:1903.05683*.

Rudolf Rosa and Zdenek Zabokrtsky. 2015. Klcpos3-a language similarity measure for delexicalized parser transfer. In *Proceedings of the 53rd ACL*, volume 2, pages 243–249.

Michael Schlichtkrull and Anders Søgaard. 2017. Cross-lingual dependency parsing with late decoding for truly low-resource languages. In *Proceedings of the 15th Conference of the EACL*, volume 1, pages 220–229.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. In *Proceedings of the NAACL*, pages 1061–1071.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the NAACL*, pages 477–487.

Jörg Tiedemann. 2015. Improving the cross-lingual projection of syntactic dependencies. In *Proceedings of the 20th NODALIDA*, 109, pages 191–199. Linköping University Electronic Press.

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 *Proceedings of the EMNLP*, pages 130–140.

Hongmin Wang, Yue Zhang, GuangYong Leonard Chan, Jie Yang, and Hai Leong Chieu. 2017. Universal dependencies parsing for colloquial singaporean english. In *Proceedings of the 55th ACL*.

Michael Wick, Pallika Kanani, and Adam Pocock. 2016. Minimally-constrained multilingual embeddings via artificial code-switching. In *AAAI*.

Min Xiao and Yuhong Guo. 2015. Annotation projection-based representation learning for cross-lingual dependency parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 73–82.

Jie Yang, Shuailong Liang, and Yue Zhang. 2018. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3879–3889.

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*.

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

Hai Zhao, Yan Song, Chunyu Kit, and Guodong Zhou. 2009. Cross language dependency parsing using a bilingual lexicon. In *Proceedings of ACL-IJCNLP*, pages 55–63.