Unsupervised Cross-Lingual Adaptation of Dependency Parsers Using CRF Autoencoders

Zhao Li and **Kewei Tu***
School of Information Science and Technology, ShanghaiTech University
Shanghai Engineering Research Center of Intelligent Vision and Imaging
{lizhao,tukw}@shanghaitech.edu.cn

Abstract

We consider the task of cross-lingual adaptation of dependency parsers without annotated target corpora and parallel corpora. Previous work either directly applies a discriminative source parser to the target language, ignoring unannotated target corpora, or employs an unsupervised generative parser that can leverage unannotated target data but has weaker representational power than discriminative parsers. In this paper, we propose to utilize unsupervised discriminative parsers based on the CRF autoencoder framework for this task. We train a source parser and use it to initialize and regularize a target parser that is trained on unannotated target data. We conduct experiments that transfer an English parser to 20 target languages. The results show that our method significantly outperforms previous methods.[1]

1 Introduction

Supervised learning of dependency parsing is difficult for low-resource languages because of the lack of large treebanks. On the other hand, cross-lingual adaptation of dependency parsers from rich-resource languages to low-resource languages has shown a lot of promise (Hwa et al., 2005; Zeman and Resnik, 2008; McDonald et al., 2011; Xiao and Guo, 2014; Tiedemann, 2015; Schlichtkrull and Søgaard, 2017; Ahmad et al., 2019), especially with the help of cross-lingual word representation (Wu and Dredze, 2019) or part-of-speech (POS) tags (Guo et al., 2015).

In this paper, we consider the scenario in which there is only unannotated data for the target language that is not parallel to the source language treebank. A simple strategy is zero-shot transfer or direct transfer, which trains a parser on the source treebank and then directly applies it to the target language (Schuster et al., 2019; Wang et al., 2019). In order to leverage unannotated target data, He et al. (2019) propose to employ an unsupervised generative parser that can be trained on the target data while also regularized via soft parameter tying by a source parser. However, generative parsers are known to underperform discriminative parsers in rich-resource scenarios, mostly because of the unrealistic independence assumptions typically made by generative parsers. In fact, He et al. (2019) show that when they use multilingual BERT (Kenton and Toutanova, 2019) as the cross-lingual word representation, their method underperforms direct transfer of a strong discriminative parser.

In this paper, we propose to instead use an unsupervised *discriminative* parser based on the CRF autoencoder framework (Ammar et al., 2014; Cai et al., 2017) for cross-lingual parser adaptation. We perform supervised training of the source parser with the source treebank and then use it to initialize the target parser. The target parser is then trained on the unannotated target data in an unsupervised way while being regularized by the source parser. We employ three regularization methods proposed by Jiang et al. (2019) that encourage similarity between model parameters and edge scores respectively of the source and target parsers. Our experiments of transferring from English to 20 target languages show that our method significantly outperforms previous methods.

2 Method

2.1 CRF Autoencoder

The CRF autoencoder is a framework of unsupervised structured prediction (Ammar et al., 2014) and has been applied to unsupervised parsing (Cai et al., 2017) and POS induction (Lin et al., 2015). It consists of an encoder that predicts a structure

*Corresponding Author

[1]Code is available at https://github.com/livc/cross-crfae.

(in our case, a dependency parse tree) from the input sentence and a decoder that reconstructs the sentence from the structure.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ be the input sentence, where x_i is the i-th word; let $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ be the dependency parse tree, where y_i is a tuple $\langle h_i, p_i \rangle$ in which h_i is the index of the dependency head of x_i and p_i is the POS tag of the head of x_i; and finally let $\hat{\mathbf{x}} = (\hat{x_1}, \hat{x_2}, \ldots, \hat{x_n})$ be the reconstructed sentence. We would like to have a perfect reconstruction, so we set $\hat{\mathbf{x}} = \mathbf{x}$.

2.1.1 Encoder

The encoder with parameters Θ computes $P_\Theta(\mathbf{y}|\mathbf{x})$. We use the deep biaffine model (Dozat and Manning, 2017), a widely used dependency parser, as our encoder. For each word x_i of the input sentence, its word and POS tag embeddings are concatenated and input into a multilayer BiLSTM to produce a contextual representation $\mathbf{r}_i$ of the word. Then $\mathbf{r}_i$ is fed into two $\mathbf{MLP}$s to produce $\mathbf{h}_i^{(dep)}$ and $\mathbf{h}_i^{(head)}$, vector representations of the word as a dependent and dependency head respectively.

We use a biaffine function to compute a score matrix $\mathbf{s}^{Enc}$, in which each element $\mathbf{s}_{i,j}^{Enc}$ is the score of the potential dependency from x_i to x_j:

$$\mathbf{s}_{i,j}^{Enc} = \mathbf{h}_i^{(\text{head})\top} W \mathbf{h}_j^{(\text{dep})} + b \tag{1}$$

where W and b are parameters of the biaffine function.

We follow the head-selection formulation of Dozat and Manning (2017) to compute $P_\Theta(\mathbf{y}|\mathbf{x})$.

$$P_\Theta(\mathbf{y}|\mathbf{x}) = \prod_i P\left(h_i|\mathbf{x}\right) \tag{2}$$

where $P\left(h_i|\mathbf{x}\right)$ can be computed by a softmax function on $\mathbf{s}^{Enc}$:

$$P\left(h_i = j|\mathbf{x}\right) = \frac{e^{\mathbf{s}_{j,i}^{Enc}}}{\sum_{k=1}^{n} e^{\mathbf{s}_{k,i}^{Enc}}} \tag{3}$$

2.1.2 Decoder

The decoder with parameters Λ computes $P_\Lambda(\hat{\mathbf{x}}|\mathbf{y})$. Following Cai et al. (2017), we represent $\hat{\mathbf{x}}$ as a sequence of POS tags instead of words and make the decoder independently predict each POS tag $\hat{p}_i$ in the reconstructed sentence conditioned only on p_i, the true POS tag of its dependency head. Our decoder simply specifies a categorical distribution $P(\hat{p}_i|p_i)$ for each possible head POS tag and computes the reconstruction probability as follows.

$$P_\Lambda(\hat{\mathbf{x}}|\mathbf{y}) = \prod_{i=1}^{n} P(\hat{p}_i|p_i) \tag{4}$$

2.1.3 Parsing

Given encoder parameters Θ and decoder parameters Λ, we can get the best parse tree by maximizing the probability $P_{\Theta,\Lambda}(\mathbf{y}, \hat{\mathbf{x}}|\mathbf{x}) = P_\Theta(\mathbf{y}|\mathbf{x}) P_\Lambda(\hat{\mathbf{x}}|\mathbf{y})$,

$$\begin{aligned}\mathbf{y}^* &= \arg\max_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} \log P_{\Theta,\Lambda}(\mathbf{y}, \hat{\mathbf{x}}|\mathbf{x}) \\ &= \arg\max_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} \sum_{i=1}^{n} (\log P\left(h_i|\mathbf{x}\right) + \log P(\hat{p}_i|p_i))\end{aligned} \tag{5}$$

where $\mathcal{Y}(\mathbf{x})$ contains all parse trees of sentence $\mathbf{x}$.

We can use Eisner's algorithm (Eisner, 1996) to find the best projective dependency parse tree in $O\left(n^3\right)$ time or use Chu-Liu/Edmonds' algorithm to find the best non-projective dependency parse tree (Chu, 1965; Edmonds, 1967; Tarjan, 1977) in $O\left(n^2\right)$ time. Additionally, we can use the head selection method (Zhang et al., 2017) in $O\left(n^2\right)$ time, which often, but not always, produce a tree structure.

2.1.4 Monolingual Learning

In the unsupervised setting, the parse tree $\mathbf{y}$ is unknown. We follow Cai et al. (2017) and minimize the negative conditional Viterbi log likelihood as the training loss function:

$$\mathcal{L} = -\sum_{i=1}^{N} \max_{\mathbf{y}\in\mathcal{Y}(\mathbf{x}_i)} \log P_{\Theta,\Lambda}\left(\hat{\mathbf{x}}_i, \mathbf{y}|\mathbf{x}_i\right) \tag{6}$$

where N is the number of training sentences. Since both the encoding and the decoding probabilities can be factorized (Eq. 2 and 4), we can rewrite Eq. 6 as follows to make it tractable.

$$\begin{aligned}\mathcal{L} &= -\sum_{i=1}^{N} \max_{\mathbf{y}\in\mathcal{Y}(\mathbf{x}_i)} \sum_{j=1}^{n_i} \left(\log P\left(h_j|\mathbf{x}_i\right) + \log P(\hat{p}_j|p_j)\right) \\ &= -\sum_{i=1}^{N} \sum_{j=1}^{n_i} \max_{y_j} \left(\log P\left(h_j|\mathbf{x}_i\right) + \log P(\hat{p}_j|p_j)\right)\end{aligned} \tag{7}$$

where n_i is the length of sentence $\mathbf{x}_i$.

In the supervised setting, the gold parse tree $\mathbf{y}^*$

is known and the loss function becomes:

$$\begin{aligned}\mathcal{L} &= -\sum_{i=1}^{N} \log P_{\Theta,\Lambda}(\hat{\mathbf{x}}_i, \mathbf{y}^*|\mathbf{x}_i) \\ &= -\sum_{i=1}^{N}\sum_{j=1}^{n_i} \left(\log P(h_j^*|\mathbf{x}_i) + \log P(\hat{p}_j|p_j^*)\right)\end{aligned} \tag{8}$$

In both settings, we can optimize encoder parameter Θ and decoder parameter Λ with stochastic gradient descent.

2.2 Cross-lingual Adaptation

To enable cross-lingual adaptation, we employ multilingual BERT (m-BERT, (Kenton and Toutanova, 2019)) and universal POS tag as the word and tag representations. We first train a CRF autoencoder (the source model) in a supervised way on the source language treebank. We then use the source model to initialize a second CRF autoencoder (the target model) and train it in an unsupervised way on the unannotated target language corpus. We stop the training after K epochs, where K is a hyper-parameter. During training of the target model, we encourage it to remain similar to the source model via regularization. We consider three forms of regularization proposed by Jiang et al. (2019).

Regularization of Model Parameters (W) The parameter regularization encourages the similarity between the source model parameters and target model parameters. Hyper-parameter λ_W controls the regularization strength. We add the following regularization term Ω to the training loss (Eq. 6).

$$\begin{aligned}\Omega = \lambda_W \ (&\|\Theta_{src} - \Theta_{tgt}\|_2^2 \\ &+ \|\Lambda_{src} - \Lambda_{tgt}\|_2^2)\end{aligned} \tag{9}$$

Regularization on Edge Scores (E) The regularization on edge scores encourages the source and target models to produce similar scores for each potential dependency in every training sentence $\mathbf{x}_i$.

$$\Omega = \lambda_E \sum_{i}^{N} \|\mathbf{s}_{src}(\mathbf{x}_i) - \mathbf{s}_{tgt}(\mathbf{x}_i)\|_2^2 \tag{10}$$

where $\mathbf{s}(\mathbf{x}_i)$ is the edge score matrix on sentence $\mathbf{x}_i$ computed by taking the summation of the encoder score $\mathbf{s}_{i,j}^{Enc}$ (Eq. 1) and the decoder score $\log P(\hat{p}_i|p_i)$ for each possible dependency edge. Hyper-parameter λ_E controls the strength of edge regularization.

Regularization on Parse Trees (T) The regularization on parse trees encourages similarity between the parse trees predicted by the source and target models. To achieve this, we change the training loss (Eq. 6) into the following form:

$$\begin{aligned}\mathcal{L} = -\sum_{i=1}^{N} \max_{\mathbf{y}\in\mathcal{Y}(\mathbf{x}_i)} \Big(&\log P_{\Theta_{tgt},\Lambda_{tgt}}(\hat{\mathbf{x}}_i, \mathbf{y}|\mathbf{x}_i) \\ &+ \lambda_T \log P_{\Theta_{src},\Lambda_{src}}(\hat{\mathbf{x}}_i, \mathbf{y}|\mathbf{x}_i)\Big)\end{aligned} \tag{11}$$

where λ_T is a hyper-parameter that controls the strength of tree regularization.

3 Experiments

3.1 Data and Setup

Our experimental setup is the same as that of He et al. (2019). We evaluate all the methods on transferring an English parser to 10 nearby languages and 10 distant languages selected from Universal Dependencies (UD) project version 2.2 (Nivre et al., 2018). We use two sets of hyper-parameters: the hyper-parameters for distant languages tuned on the Arabic development set and the hyper-parameters for nearby languages tuned on the Spanish development set.

For supervised learning of the source model, we train on sentences of all lengths. For unsupervised learning of the target model, we train on sentences of length ≤ 40. We test the target model on sentences of all lengths and use Eisner's algorithm for parsing.

We run each experiment for five times with different random seeds on a Tesla P40 GPU and report the average unlabeled attachment score (UAS) with punctuation excluded.

3.2 Results

We compare our method with a previous state-of-the-art approach (He et al., 2019) and several baselines in Table 1. The three generative methods are from He et al. (2019): **F-Fix** is their Flow-Fix model that directly transfers the generative source model, **F-N** is their Flow-FT model that trains on the target corpus without source regularization, and **F-FT** is their best-performing Flow-FT model that trains on the target corpus with source regularization. We rerun their source code[2] in our experiments. For discriminative models, **DT** is the direct transfer baseline and **S-T** is the self-training

[2] https://github.com/jxhe/cross-lingual-struct-flow

	Generative (He et al., 2019)			Discriminative									
Lang	**F-Fix**	**F-N**	**F-FT**	**DT**	**S-T**	**Fix**	**N**	**W**	**E**	**T**	**W+E**	**W+T**	**E+T**
Distant Languages													
zh (0.86)	36.05	24.14	26.27	57.49	**59.83**	54.62	44.77	45.00	45.47	45.50	45.13	45.41	46.97
fa (0.86)	36.79	46.68	58.33	49.46	51.38	50.67	**61.98**	61.45	61.14	59.89	60.02	60.55	60.43
ar (0.86)	31.86	54.86	54.97	43.86	41.90	45.66	65.27	**65.84**	64.96	64.88	64.89	64.72	64.22
ja (0.71)	19.59	37.08	42.45	35.40	36.68	40.41	62.91	62.55	**64.15**	63.34	64.07	62.75	63.08
id (0.71)	48.73	50.88	66.31	53.43	52.44	54.18	62.69	63.78	**64.03**	63.14	63.86	63.10	61.21
ko (0.69)	32.93	37.82	33.84	45.62	**47.34**	47.02	34.74	36.14	34.18	34.20	33.50	34.93	42.08
tr (0.62)	36.95	32.51	34.16	48.84	50.57	49.40	51.23	51.54	51.43	51.51	51.49	**51.56**	51.39
hi (0.61)	28.70	21.94	31.96	50.67	52.28	53.63	56.54	55.97	58.59	58.26	57.22	59.44	**59.87**
hr (0.59)	59.48	48.06	64.29	79.61	78.32	78.77	83.45	83.31	82.93	83.16	**84.14**	83.65	83.92
he (0.57)	52.15	56.14	64.74	66.49	65.61	66.52	73.44	**73.85**	72.99	72.89	73.30	72.51	73.30
AVG	38.32	41.01	47.73	53.09	53.76	54.09	59.70	59.94	59.99	59.68	59.76	59.86	**60.65**
Nearby Languages													
bg (0.50)	70.74	50.74	71.40	88.66	**88.96**	88.76	87.68	88.65	87.62	88.43	88.21	88.23	88.13
it (0.50)	69.17	53.24	70.98	84.96	85.56	85.63	90.17	90.58	89.65	89.91	**90.75**	90.67	89.59
pt (0.48)	66.95	47.82	66.70	79.98	80.56	80.48	84.62	**86.35**	84.40	84.69	85.97	86.01	84.19
fr (0.46)	66.89	47.72	67.94	82.89	83.26	83.30	86.09	87.35	86.19	86.19	**87.74**	87.68	86.92
es (0.46)	64.02	47.12	64.70	79.14	79.45	79.70	80.70	**84.32**	80.87	81.40	84.21	84.19	81.62
no (0.45)	64.61	45.24	64.17	86.74	**87.19**	86.71	78.76	86.88	81.52	81.88	86.84	86.87	83.14
da (0.41)	61.41	41.76	60.71	83.27	83.30	**83.63**	82.35	83.27	82.57	82.50	83.19	83.26	82.74
sv (0.40)	65.25	47.51	63.83	86.27	85.97	86.74	87.05	**87.09**	86.74	86.98	86.89	86.89	86.83
nl (0.37)	61.54	34.74	61.78	79.59	80.91	79.76	79.25	**81.05**	80.05	80.11	80.76	80.83	80.57
de (0.36)	65.88	36.95	65.25	79.39	80.70	80.91	84.12	84.93	85.06	84.96	84.69	84.70	**85.43**
AVG	65.64	45.28	65.75	83.09	83.59	83.56	84.08	**86.05**	84.47	84.71	85.93	85.93	84.92
en*	66.94	–	–	92.70	–	92.49	–	–	–	–	–	–	–

Table 1: Dependency parsing results (UAS %) on target languages. Numbers next to language names are their distances to English copied from He et al. (2019). Supervised results on English (∗) are included for reference.

baseline, both of which use the biaffine parser (Dozat and Manning, 2017). **S-T** follows Rybak and Wróblewska (2018) who use the source model to predict parse trees on the target data and then perform supervised training of the target model. The last eight methods are our methods. **Fix** is direct transfer of the CRF autoencoder. **N** is our method without any regularization. **W**, **E** and **T** are our method with weight, edge and tree regularization respectively. **W+E**, **W+T** and **E+T** are our method with two forms of regularization combined.

As shown in Table 1, all the discriminative methods outperform the three generative methods on average, and the performance gap is especially large on nearby languages. This is consistent with the findings of He et al. (2019) when using m-BERT.

Comparing the discriminative methods, we find that our methods clearly outperform the **DT**, **S-T** and **Fix** baselines on both distant languages and nearby languages, showing the advantage of unsupervised training on target data. However, the improvements produced by our methods on nearby languages are much smaller than those on distant languages. This is not surprising considering that nearby languages share similar syntactic behaviors and direct transfer can already produce strong parsers.

Comparing our methods with and without regularization, we see that regularization helps in most cases. The usefulness of regularization is more prominent on nearby languages, probably because of the better performance of the source model on nearby languages.

3.3 Analysis

We evaluate our model with varying sizes of the target/source data and fixed source/target data in Figure 1. It can be seen that more target data can boost the accuracy on the distant language (Arabic), but hurt the accuracy on the nearby language (Spanish) unless alleviated by regularization. On the other hand, more source data is always helpful, especially on the distant language.

4 Conclusion

In this paper, we employ unsupervised discriminative parsers based on the CRF autoencoder framework for unsupervised cross-lingual adaptation of dependency parsers. We initialize the target model using the source model and train it on unannotated target data in an unsupervised way, with three forms of regularization that encourage its similarity

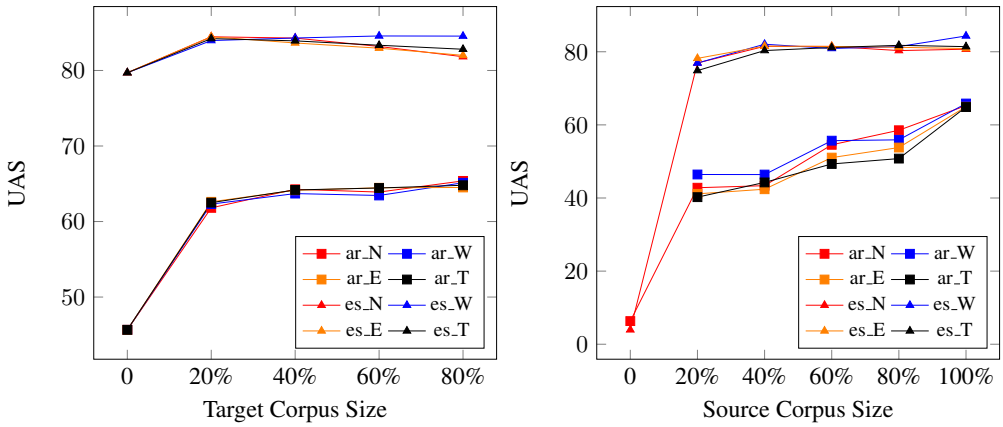

Figure 1: Dependency parsing results (UAS %) with varying corpus sizes. Left: fixed source corpus size and varying target corpus size. Right: fixed target corpus size and varying source corpus size (0: no source corpus and hence no source model, so the target model is obtained solely by unsupervised learning). ar: Arabic. es: Spanish.

to the source model. Our experiments show the advantage of our methods over previous generative methods and discriminative baselines.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61976139).

References

Wasi 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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2440–2452.

Waleed Ammar, Chris Dyer, and Noah A Smith. 2014. Conditional random field autoencoders for unsupervised structured prediction. In *Advances in Neural Information Processing Systems*, pages 3311–3319.

Jiong Cai, Yong Jiang, and Kewei Tu. 2017. Crf autoencoder for unsupervised dependency parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1638–1643.

Yoeng-Jin Chu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400.

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*.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240.

Jason M. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *COLING 1996 Volume 1: The 16th International Conference on 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.

Junxian He, Zhisong Zhang, Taylor Berg-Kirkpatrick, and Graham Neubig. 2019. Cross-lingual syntactic transfer through unsupervised adaptation of invertible projections. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3211–3223.

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.

Yong Jiang, Wenjuan Han, and Kewei Tu. 2019. A regularization-based framework for bilingual grammar induction. 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 1423–1428.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Chu-Cheng Lin, Waleed Ammar, Chris Dyer, and Lori Levin. 2015. Unsupervised pos induction with word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1311–1316.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 62–72.

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.

Piotr Rybak and Alina Wróblewska. 2018. Semi-supervised neural system for tagging, parsing and lemmatization. *CoNLL 2018*, page 45.

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 European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 220–229.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. 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 1599–1613.

Robert Endre Tarjan. 1977. Finding optimum branchings. *Networks*, 7(1):25–35.

Jörg Tiedemann. 2015. Cross-lingual dependency parsing with universal dependencies and predicted pos labels. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 340–349.

Yuxuan Wang, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu. 2019. Cross-lingual BERT transformation for zero-shot 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 5721–5727.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. 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 833–844.

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.

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*.

Xingxing Zhang, Jianpeng Cheng, and Mirella Lapata. 2017. Dependency parsing as head selection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 665–676.

A Model Hyperparameters

Parameter	Description	DT Baseline Value	CRF Autoencoder Value
word_embed	dimension of word embeddings	300	300
n_embed	dimension of pos tag embeddings	300	150
n_bert_layers	number of bert layers to use	4	4
embed_dropout	dropout ratio of embeddings	0.33	0.33
n_lstm_hidden	dimension of lstm hidden states	400	200
n_lstm_layers	number of lstm layers	3	3
lstm_dropout	dropout ratio of lstm	0.33	0.33
n_mlp_arc	arc mlp size	500	50
mlp_dropout	dropout ratio of mlp	0.33	0.33
lr	starting learning rate of training	2e-3	1e-3
betas	hyperparameters of momentum and L2 norm	(0.9, 0.9)	(0.9, 0.9)
epsilon	stability constant	1e-12	1e-12
K	unsupervised training epoch	-	1

B Regularization Parameters

The regularization parameters are tuned on the development set of Arabic for distant languages and Spanish for nearby languages.

		distant	nearby
W		1e6	1e8
E		1e-12	1e-8
T		1e-6	1e-2
W+E	W	1e-8	1e8
	E	1e-10	1e-12
W+T	W	1e-4	1e8
	T	1e-4	1e-4
E+T	E	1e-10	1e-8
	T	1e-2	1e-2