Cross-Lingual Syntactic Transfer through Unsupervised Adaptation of Invertible Projections

Junxian He[1], **Zhisong Zhang**[1], **Taylor Berg-Kirkpatrick**[2], **and Graham Neubig**[1]
[1]Language Technologies Institute, Carnegie Mellon University
[2]Department of Computer Science and Engineering, University of California San Diego
{junxianh,zhisongz,gneubig}@cs.cmu.edu, tberg@eng.ucsd.edu

Abstract

Cross-lingual transfer is an effective way to build syntactic analysis tools in low-resource languages. However, transfer is difficult when transferring to typologically distant languages, especially when neither annotated target data nor parallel corpora are available. In this paper, we focus on methods for cross-lingual transfer to distant languages and propose to learn a generative model with a structured prior that utilizes labeled source data and unlabeled target data jointly. The parameters of source model and target model are softly shared through a regularized log likelihood objective. An invertible projection is employed to learn a new interlingual latent embedding space that compensates for imperfect cross-lingual word embedding input. We evaluate our method on two syntactic tasks: part-of-speech (POS) tagging and dependency parsing. On the Universal Dependency Treebanks, we use English as the only source corpus and transfer to a wide range of target languages. On the 10 languages in this dataset that are distant from English, our method yields an average of 5.2% absolute improvement on POS tagging and 8.3% absolute improvement on dependency parsing over a direct transfer method using state-of-the-art discriminative models.[1]

1 Introduction

Current top performing systems on syntactic analysis tasks such as part-of-speech (POS) tagging and dependency parsing rely heavily on large-scale annotated data (Huang et al., 2015; Dozat and Manning, 2017; Ma et al., 2018). However, because creating syntactic treebanks is an expensive and time consuming task, annotated data is scarce for many languages. Prior work has demonstrated the efficacy of cross-lingual learning methods (Guo et al., 2015; Tiedemann, 2015; Guo et al., 2016; Zhang et al., 2016; Ammar et al., 2016; Ahmad et al., 2019; Schuster et al., 2019), which transfer models between different languages through the use of shared features such as cross-lingual word embeddings (Smith et al., 2017; Conneau et al., 2018) or universal part-of-speech tags (Petrov et al., 2012). In the case of *zero-shot* transfer (i.e. with no target-side supervision), a common practice is to train a strong supervised system on the source language and directly apply it to the target language over these shared embedding or POS spaces. This method has demonstrated promising results, particularly for transfer of models to closely related target languages (Ahmad et al., 2019; Schuster et al., 2019).

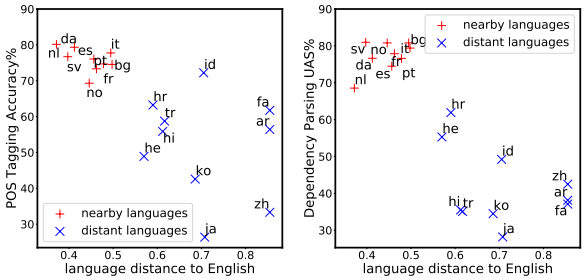

Figure 1: **Left:** POS tagging transfer accuracy of the Bidirectional LSTM-CRF model, **Right:** Dependency parsing transfer UAS of the "SelfAtt-Graph" model (Ahmad et al., 2019). These models are trained on the labeled English corpus and directly evaluated on different target languages. The x-axis represents language distance to English (details in Section 2.1). Both models take pre-trained cross-lingual word embeddings as input. The parsing model also uses gold universal POS tags.

However, this direct transfer approach often produces poor performance when transferring to more distant languages that are less similar to the source. For example, in Figure 1 we show

[1]Code is available at https://github.com/jxhe/cross-lingual-struct-flow.

the results of direct transfer of POS taggers and dependency parsers trained on only English and evaluated on 20 target languages using pre-trained cross-lingual word embeddings, where the x-axis shows the linguistic distance from English calculated according to the URIEL linguistic database (Littell et al., 2017) (more details in Section 2). As we can see, these systems suffer from a large performance drop when applied to distant languages. The reasons are two-fold: (1) Cross-lingual word embeddings of distant language pairs are often poorly aligned with current methods that make strong assumptions of orthogonality of embedding spaces (Smith et al., 2017; Conneau et al., 2018). (2) Divergent syntactic characteristics make the model trained on the source language non-ideal, even if the cross-lingual word embeddings are of high quality.

In this paper we take a drastically different approach from most previous work: instead of directly transferring a *discriminative* model trained only on labeled data in another language, we use a *generative* model that can be trained in an supervised fashion on labeled data in another language, but also perform unsupervised training to directly maximize likelihood of the target language. This makes it possible to specifically adapt to the language that we would like to analyze, both with respect to the cross-lingual word embeddings and the syntactic parameters of the model itself.

Specifically, our approach builds on two previous works. We follow a training strategy similar to Zhang et al. (2016), who have previously demonstrated that it is possible to do this sort of cross-lingual unsupervised adaptation, although limited to the sort of linear projections that we argue are too simple for mapping between embeddings in distant languages. To relax this limitation, we follow He et al. (2018) who, in the context of fully unsupervised learning, propose a method using invertible projections (which is also called *flow*) to learn more expressive transformation functions while nonetheless maintaining the ability to train in an unsupervised manner to maximize likelihood. We learn this structured flow model (detailed in Section 3.1) on both labeled source data and unlabeled target data through a soft parameter sharing scheme. We describe how to apply this method to two syntactic analysis tasks: POS tagging with a hidden Markov model (HMM) prior and dependency parsing with a dependency model with valence (DMV; Klein and Manning (2004)) prior (Section 4.3).

Language Category	Language Names
Distant	Chinese (zh, 0.86), Persian (fa, 0.86), Arabic (ar, 0.86), Japanese (ja, 0.71), Indonesian (id, 0.71), Korean (ko, 0.69), Turkish (tr, 0.62), Hindi (hi, 0.61), Croatian (hr, 0.59), Hebrew (he, 0.57)
Nearby	Bulgarian (bg, 0.50), Italian (it, 0.50), Portuguese (pt, 0.48), French (fr, 0.46), Spanish (es, 0.46), Norwegian (no, 0.45) Danish (da, 0.41), Swedish (sv, 0.40) Dutch (nl, 0.37), German (de, 0.36)

Table 1: 20 selected target languages. Numbers in the parenthesis denote the distances to English.

We evaluate our method on Universal Dependencies Treebanks (v2.2) (Nivre et al., 2018), where English is used as the only labeled source data. 10 distant languages and 10 nearby languages are selected as the target without labels. On 10 distant transfer cases – which we focus on in this paper – our approach achieves an average of 5.2% absolute improvement on POS tagging and 8.3% absolute improvement on dependency parsing over strong discriminative baselines. We also analyze the performance difference between different systems as a function of language distance, and provide preliminary guidance on when to use generative models for cross-lingual transfer.

2 Difficulties of Cross-Lingual Transfer on Distant Languages

In this section, we demonstrate the difficulties involved in performing cross-lingual transfer to distant languages. Specficially, we investigate the direct transfer performance as a function of language distances by training a high-performing system on English and then apply it to target languages. We first introduce the measurement of language distances and selection of 20 target languages, then study the transfer performance change on POS tagging and dependency parsing tasks.

2.1 Language Distance

To quantify language distances, we make use of the URIEL (Littell et al., 2017) database,[2] which represents over 8,000 languages as information-rich typological, phylogenetic, and geographical vectors. These vectors are sourced and predicted

[2] http://www.cs.cmu.edu/~dmortens/uriel.html

from a variety of linguistic resources such as WALS (Dryer, 2013), PHOIBLE (Moran et al., 2014), Ethnologue (Lewis et al., 2015), and Glottolog (Hammarstrm et al., 2015). Based on these vectors, this database provides ready-to-use distance statistics between any pair of languages included in the database in terms of various metrics including *genetic* distance, *geographical* distance, *syntactic* distance, *phonological* distance, and phonetic *inventory* distance. These distances are represented by values between 0 and 1. Since phonological and inventory distances mainly characterize intra-word phonetic/phonological features that have less effect on word-level language composition rules, we remove these two and take the average of genetic, geographic, and syntactic distances as our distance measure.

We rank all languages in Universal Dependencies (UD) Treebanks (v2.2) (Nivre et al., 2018) according to their distances to English, with the distant ones on the top. Then we select 10 languages from the top that represent the *distant language* group, and 10 languages from the bottom that represent the *nearby language* group. The selected languages are required to meet the following two conditions: (1) at least 1,000 unlabeled training sentences present in the treebank since a reasonably large amount of unlabeled data is needed to study the effect of unsupervised adaptation, and (2) an offline pre-trained word embedding alignment matrix is available.[3] The 20 selected target languages are shown in Table 1, which contains distant languages like Persian and Arabic, but also closely related languages like Spanish and French. Detailed statistical information of the selected languages and corresponding treebanks can be found in Appendix A.

2.2 Observations

In the direct transfer experiments, we use the pre-trained cross-lingual fastText word embeddings (Bojanowski et al., 2017), aligned with the method of Smith et al. (2017). These embeddings are fixed during training otherwise the alignment would be broken. We employ a bidirectional LSTM-CRF (Huang et al., 2015) model for POS tagging using NCRF++ toolkit (Yang and Zhang, 2018), and use the "SelfAtt-Graph" model (Ahmad et al., 2019) for dependency parsing.[4] Following Ahmad et al. (2019), for dependency parsing gold POS tags are also used to learn POS tag embeddings as universal features. We train the systems on English and directly evaluate them on the target languages. Results are shown in Figure 1. While these systems achieve quite accurate results on closely related languages, we observe large performance drops on both tasks as distance to English increases. These results motivate our proposed approach, which aims to close this gap by directly adapting to the target language through unsupervised learning over unlabeled text.

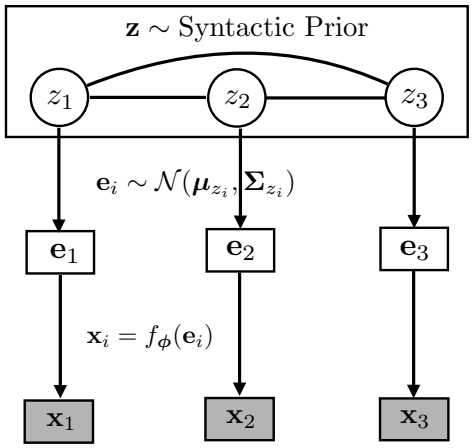

Figure 2: Graphical representation of the structured flow model. We denote discrete syntactic variables as $\mathbf{z}$, latent embedding variable as $\mathbf{e}$, and observed pre-trained word embeddings as $\mathbf{x}$. f_{ϕ} is the invertible projection function.

3 Proposed Method

In this section, we first introduce the unsupervised monolingual models presented in He et al. (2018), which we refer to as *structured flow models*, then we propose our approach that extends the structured flow models to cross-lingual settings.

3.1 Unsupervised Training of Structured Flow Models

The structured flow generative model, proposed by He et al. (2018), is a state-of-the-art technique for inducing syntactic structure in a monolingual setting without supervision. This model cascades a structured generative prior $p_{\text{syntax}}(\mathbf{z}; \boldsymbol{\theta})$ with an invertible neural network $f_{\phi}(\mathbf{z})$ to generate pre-

[3] Following Ahmad et al. (2019), we use the offline pre-trained alignment matrix present in https://github.com/Babylonpartners/fastText_multilingual, which contains alignment matrices for 78 languages, which also allows comparison with their numbers in Section 4.3.

[4] We use an implementation and English source model checkpoint identical to the original paper.

trained word embeddings $\mathbf{x} = f_{\phi}(\mathbf{z})$, which correspond to the words in the training sentences. $\mathbf{z}$ represents latent syntax variables that are not observed during training. The structured prior defines a probability over syntactic structures, and can be a Markov prior to induce POS tags or DMV prior (Klein and Manning, 2004) to induce dependency structures. Notably, the model side-steps discrete words, and instead uses pre-trained word embeddings as observations, which allows it to be directly employed in cross-lingual transfer setting by using cross-lingual word embeddings as the observations. A graphical illustration of this model is shown in Figure 2. Given a sentence of length l, we denote $\mathbf{z} = \{z\}_{k=1}^{K}$ as a set of discrete latent variables from the structured prior, $\mathbf{e} = \{\mathbf{e}_i\}_{i=1}^{l}$ as the latent embeddings, and $\mathbf{x} = \{\mathbf{x}_i\}_{i=1}^{l}$ as the observed word embeddings. Note that the number of latent syntax variables K is no smaller than the sentence length l, and we assume $\mathbf{x}_i$ is generated (indirectly) conditioned on $\mathbf{z}_i$ for notation simplicity. The model is trained by maximizing the following marginal data likelihood:

$$p_{\text{us}}(\mathbf{x}) = \sum\nolimits_{\mathbf{z}} \Big(p_{\text{syntax}}(\mathbf{z}; \boldsymbol{\theta}) \cdot \prod\nolimits_{i=1}^{\ell} p_{\boldsymbol{\eta}}(f_{\boldsymbol{\phi}}^{-1}(\mathbf{x}_i)|z_i) \Big| \det \frac{\partial f_{\boldsymbol{\phi}}^{-1}}{\partial \mathbf{x}_i} \Big| \Big). \quad (1)$$

$p_{\boldsymbol{\eta}}(\cdot|z_i)$ is defined to be a conditional Gaussian distribution that emits latent embedding $\mathbf{e}$. The projection function $f_{\boldsymbol{\phi}}$ projects the latent embedding $\mathbf{e}$ to the observed embedding $\mathbf{x}$. $\frac{\partial f_{\boldsymbol{\phi}}^{-1}}{\partial \mathbf{x}_i}$ is the Jacobian matrix of function $f_{\boldsymbol{\phi}}^{-1}$ at $\mathbf{x}_i$, and $\big|\det \frac{\partial f_{\boldsymbol{\phi}}^{-1}}{\partial \mathbf{x}_i}\big|$ represents the absolute value of its determinant.

To understand the intuitions behind Eq. 1, first denote the log likelihood over the latent embedding $\mathbf{e}$ as $\log p_{\text{gaus}}(\cdot)$, then log of Eq. 1 can be equivalently rewritten as:

$$\log p_{\text{us}}(\mathbf{x}) = \log p_{\text{gaus}}(f_{\boldsymbol{\phi}}^{-1}(\mathbf{x})) + \sum\nolimits_{i=1}^{l} \log \Big| \det \frac{\partial f_{\boldsymbol{\phi}}^{-1}}{\partial \mathbf{x}_i} \Big|. \quad (2)$$

Eq. 2 shows that $f_{\phi}^{-1}(\mathbf{x})$ inversely projects $\mathbf{x}$ to a new latent embedding space, on which the unsupervised training objective is simply the Gaussian log likelihood with an additional Jacobian regularization term. The Jacobian regularization term accounts for the volume expansion or contraction behavior of the projection, thus maximizing it can be thought of as preventing information loss.[5] This projection scheme can flexibly transform embedding space to fit the task at hand, but still avoids trivial solutions by preserving information.

While $f_{\boldsymbol{\phi}}^{-1}(\mathbf{x})$ can be any invertible function, He et al. (2018) use a version of the NICE architecture (Dinh et al., 2014) to construct $f_{\boldsymbol{\phi}}^{-1}$, which has the advantage that the determinant term is constantly equal to one. This structured flow model allows for exact marginal data likelihood computation and exact inference by the use of dynamic programs to marginalize out $\mathbf{z}$. More details about this model can be found in He et al. (2018).

3.2 Supervised Training of Structured Flow Models

While He et al. (2018) train the structured flow model in an unsupervised fashion, this model can be also trained with supervised data when $\mathbf{z}$ is observed. Supervised training is required in the cross-lingual transfer where we train a model on the high-resource source language. The supervised objective can be written as:

$$p_{\text{s}}(\mathbf{z}, \mathbf{x}) = p_{\text{syntax}}(\mathbf{z}; \boldsymbol{\theta}) \cdot \prod\nolimits_{i=1}^{\ell} p_{\boldsymbol{\eta}}(f_{\boldsymbol{\phi}}^{-1}(\mathbf{x}_i)|z_i) \Big| \det \frac{\partial f_{\boldsymbol{\phi}}^{-1}}{\partial \mathbf{x}_i} \Big|, \quad (3)$$

3.3 Multilingual Training through Parameter Sharing

In this paper, we focus on the zero-shot cross-lingual transfer setting where the syntactic structure $\mathbf{z}$ is observed for the source language but unavailable for the target languages. Eq. 2 is an unsupervised objective which is optimized on the target languages, and Eq. 3 is optimized on the source language. To establish connections between the source and target languages, we employ two instances of the structured flow model – a *source model* and a *target model* – and share parameters between them. The source model uses the supervised objective, Eq. 3, and the target model uses the unsupervised objective, Eq. 2, and both are optimized jointly. Instead of tying their parameters in a hard way, we share their parameters softly through an L2 regularizer that encourages similarity. We use subscript p to represent variables of the source model and q to represent variables of

[5] Maximizing the Jacobian term encourages volume expansion and prevents the latent embedding from collapsing to a (nearly) single point.

the target model. Together, our joint training objective is:

$$L(\boldsymbol{\theta}_{\{p,q\}}, \boldsymbol{\eta}_{\{p,q\}}, \boldsymbol{\phi}_{\{p,q\}}) = \log p_{\text{s}}(\mathbf{x}_p) + \log p_{\text{us}}(\mathbf{x}_q) - \frac{\beta_1}{2}\|\boldsymbol{\theta}_p - \boldsymbol{\theta}_q\|^2 - \frac{\beta_2}{2}\|\boldsymbol{\eta}_p - \boldsymbol{\eta}_q\|^2 - \frac{\beta_3}{2}\|\boldsymbol{\phi}_p - \boldsymbol{\phi}_q\|^2, \quad (4)$$

where $\boldsymbol{\beta} = \{\beta_1, \beta_2, \beta_3\}$ are regularization parameters. Introduction of hyperparameters is concerning because in the zero-shot transfer setting we do not have annotated data to select the parameters for each target language, but in experiments we found it unnecessary to tune $\boldsymbol{\beta}$ for different target languages separately, and it is possible to use the same $\boldsymbol{\beta}$ within the same language category (i.e. distant or nearby). Under the parameter sharing scheme the projected latent embedding space $\mathbf{e}$ can be understood as the new interlingual embedding space from which we learn the syntactic structures. The expressivity of the flow model used in learning this latent embeddings space is expected to compensate for the imperfect orthogonality between the two embedding spaces.

Further, jointly training both models with Eq. 4 is more expensive than typical cross-lingual transfer setups – it would require re-training both models for each language pair. To improve efficiency and memory utilization, in practice we use a simple pipelined approach: (1) We pre-train parameters for the source model only once, in isolation. (2) We use these parameters to initialize each target model, and regularize all target parameters towards this initializer via the L2 terms in Eq. 4. In this way, we only need to save the pre-trained parameters for a single source model, and target-side fine-tuning converges much faster than training each pair from scratch. This training approximation has been used before in Zhang et al. (2016).

4 Experiments

In this section, we first describe the dataset and experimental setup, and then report the cross-lingual transfer results of POS tagging and dependency parsing on distant target languages. Lastly we include analysis of different systems.

4.1 Experimental Setup

Across both POS tagging and dependency parsing tasks, we run experiments on Universal Dependency Treebanks (v2.2) (Nivre et al., 2018). Specifically, we train the proposed model on the English corpus with annotated data and fine-tune it on target languages in an unsupervised way. In the rest of the paper we will use *Flow-FT* to term our proposed method. We use the aligned cross-lingual word embeddings described in Section 2.2 as the observations of our model. To compare with Ahmad et al. (2019), on dependency parsing task we also use universal gold POS tags to index tag embeddings as part of observations. Specifically, the tag embeddings are concatenated with word embeddings to form $\mathbf{x}$, tag embeddings are updated when training on the source language, and fixed at fine-tuning stage. We implement the structured flow model based on the public code from He et al. (2018),[6]which contains models with Markov prior for POS tagging and DMV prior for dependency parsing. Detailed hyperparameters can be found in Appendix B. Both source model and target model are optimized with Adam (Kingma and Ba, 2014). Training on the English source corpus is run 5 times with different random restarts for all models, then the source model with the best English test accuracy is selected to perform transfer.

We compare our method with a direct transfer approach that is based on the state-of-the-art discriminative models as described in Section 2.2. The pre-trained cross-lingual word embeddings for all models are frozen since fine-tuning them will break the multi-lingual alignments. In addition, to demonstrate the efficacy of unsupervised adaptation, we also include direct transfer results of our model without fine-tuning, which we denote as *Flow-Fix*. On the POS tagging task we re-implement the generative baseline in Zhang et al. (2016) that employs a linear projection (*Linear-FT*). We present results on 20 target languages in "distant languages" and "nearby languages" categories to analyze the difference of the systems and the scenarios to which they are applicable.

4.2 Part-Of-Speech Tagging

Setup. Our method aims to predict coarse universal POS tags, as fine-grained tags are language-dependent. The discriminative baseline with the NCRF++ toolkit (Yang and Zhang, 2018) achieves supervised test accuracy on English of 94.02%, which is competitive (rank 12) on the CoNLL

[6] https://github.com/jxhe/struct-learning-with-flow.

	Discriminative	Generative		
Lang	**LSTM-CRF**	**Flow-Fix**	**Flow-FT**	**Linear-FT**
Distant Languages				
zh (0.86)	33.31	35.24	**43.44**	35.95
fa (0.86)	61.74	55.32	**64.47**	34.35
ar (0.86)	56.41	49.70	**64.00**	38.95
ja (0.71)	26.37	25.09	**38.37**	12.49
id (0.71)	72.21	63.73	**73.51**	57.56
ko (0.69)	**42.57**	39.56	41.76	18.30
tr (0.62)	58.74	43.17	**60.08**	22.79
hi (0.61)	55.85	47.18	**64.75**	38.04
hr (0.59)	**63.23**	50.57	57.90	56.53
he (0.57)	48.90	47.97	**62.69**	48.17
AVG	51.93	45.75	**57.10**	36.31
Nearby Languages				
bg (0.50)	**74.55**	62.18	64.69	66.71
it (0.50)	77.75	69.93	**80.99**	73.55
pt (0.48)	**74.68**	65.08	72.65	72.54
fr (0.46)	**73.33**	64.15	69.78	66.63
es (0.46)	76.07	65.77	**77.19**	72.86
no (0.45)	**69.30**	58.98	62.05	62.38
da (0.41)	**79.33**	62.42	68.68	67.31
sv (0.40)	**76.70**	58.91	66.34	61.82
nl (0.37)	**80.15**	66.52	68.74	66.08
de (0.36)	**68.75**	57.91	59.97	56.16
AVG	**75.06**	63.19	69.11	66.60
en*	94.02	87.03	–	84.69

Table 2: POS tagging accuracy results (%). Numbers next to languages names are their distances to English. Supervised accuracy on English (∗) is included for reference.

2018 Shared Task scoreboard that uses the same dataset.[7] The regularization parameters $\boldsymbol{\beta}$ in all generative models are tuned on the Arabic[8] development data and kept the same for all target languages. Our running $\boldsymbol{\beta}$ is $\beta_1 = 0, \beta_2 = 500, \beta_3 = 80$. Unsupervised fine-tuning is run for 10 epochs.

Results. We show our results in Table 2, where unsupervised fine-tuning achieves considerable and consistent performance improvements over the Flow-Fix baseline in both language categories. When compared the discriminative LSTM-CRF baseline, our approach outperforms it on 8 out of 10 distant languages, with an average of 5.2% absolute improvement. Unsurprisingly, however, it also underperforms the expressive LSTM-CRF on 8 out of 10 nearby languages. The reasons for this phenomenon are two-fold. First, the flexible LSTM-CRF model is better able to fit the source English corpus than our method (94.02% vs 87.03% accuracy), thus it is also capable of fitting similar input when transferring. Second, unsupervised adaptation helps less when transferring to nearby languages (5.9% improvement over Flow-Fix versus 11.3% on distant languages), we posit that this is because a large portion of linguistic knowledge is shared between similar languages, and the cross-lingual word embeddings have better quality in this case, so unsupervised adaptation becomes less necessary. While the Linear-FT baseline on nearby languages is comparable to our method, its performance on distant languages is much worse, which confirms the importance of invertible projection, especially when language typologies are divergent.

4.3 Dependency Parsing

Setup. In preliminary parsing results we found that transferring to "nearby language" group is likely to suffer from catastrophic forgetting (McCloskey and Cohen, 1989) and thus requires stronger regularization towards the source model. This also makes sense intuitively since nearby languages should prefer the source model more than distant languages. Therefore, we use two different sets of regularization parameters for nearby languages and distant languages, respectively. Specifically, $\boldsymbol{\beta}$ for the "distant languages" group is set as $\beta_1 = \beta_2 = \beta_3 = 0.1$, tuned on the Arabic development set, and for the "nearby languages" group $\boldsymbol{\beta}$ is set as $\beta_1 = \beta_2 = \beta_3 = 1$, tuned on the Spanish development set. Unsupervised adaptation is performed on sentences of length less than 40 due to memory constraints,[9] but we test on sentences of all lengths. We run unsupervised fine-tuning for 5 epochs, and evaluate using unlabeled attachment score (UAS) with punctuation excluded.

Results. We show our results in Table 3. While unsupervised fine-tuning improves the performance on the distant languages, it only has minimal effect on nearby languages, which is consistent with our observations in the POS tagging experiment and implies that unsupervised adaption helps more for distant transfer. Similar to POS tagging results, our method is able to outperform state-of-the-art "SelfAtt-Graph" model on 8 out of 10 distant languages, with an average of 8.3%

[7] For reference, check the "en_ewt" treebank results in http://universaldependencies.org/conll18/results-upos.html.

[8] We choose Arabic simply because it is first in alphabetical order.

[9] Reducing batch size can address this memory issue, but greatly increases the training time.

	Discriminative	Generative	
Lang	**SelfAtt-Graph**	**Flow-Fix**	**Flow-FT**
Distant Languages			
zh (0.86)	**42.48**	35.72	37.26
fa (0.86)	37.10	37.58	**63.20**
ar (0.86)	38.12	32.14	**55.44**
ja (0.71)	28.18	19.03	**43.75**
id (0.71)	49.20	46.74	**64.20**
ko (0.69)	34.48	34.76	**37.03**
tr (0.62)	35.08	34.76	**36.05**
hi (0.61)	**35.50**	29.20	33.17
hr (0.59)	61.91	59.57	**65.31**
he (0.57)	55.29	51.35	**64.80**
AVG	41.73	38.09	**50.02**
Nearby Languages			
bg (0.50)	**79.40**	73.52	73.57
it (0.50)	**80.80**	68.84	70.68
pt (0.48)	**76.61**	66.61	66.61
fr (0.46)	**77.87**	65.92	67.66
es (0.46)	**74.49**	63.10	64.28
no (0.45)	**80.80**	65.48	65.29
da (0.41)	**76.64**	61.64	61.08
sv (0.40)	**80.98**	66.22	64.43
nl (0.37)	**68.55**	61.59	61.72
de (0.36)	**71.34**	70.10	69.52
AVG	**76.75**	66.30	66.48
en*	91.82	67.80	–

Table 3: Dependency parsing UAS (%) on sentences of all lengths. Numbers next to languages names are their distances to English. Supervised accuracy on English (∗) is included for reference.

absolute improvement, but the strong discriminative baseline performs better when transferring to nearby languages. Note that the supervised performance of our method on English is poor. This is mainly because the DMV prior is too simple and limits the capacity of the model. While this model still achieves good performance on distant transfer, incorporating more complex DMV variants (Jiang et al., 2016) might lead to further improvement.

Analysis on Dependency Relations. We further perform breakdown analysis on dependency relations to see how unsupervised adaptation helps learn new dependency rules. We select three typical distant languages with different word order of Subject, Object and Verb (Dryer, 2013): Arabic (Modern Standard, VSO), Indonesian (SVO) and Japanese (SOV).

We investigate the unlabeled accuracy (recall) on the gold dependency labels. We especially explore four typical dependency relations: case (case marking), nmod (nominal modifier), obj (object) and nsubj (nominal subject). The first two are "nominal dependents" (modifiers for nouns) and the rest two are the main nominal "core arguments" (arguments for the predicate). Although different languages may vary, these four types are representative relations and occupies 25% to 40% in frequencies among all 37 UD dependency types.

We compare our fine-tuning model with the baseline "SelfAtt-Graph" model and our basic model without fine-tuning. As shown in Figure 3, although our direct transfer model obtain similar results when compared with the baseline, the fine-tuning method brings large improvements on most of these dependency relations. In these three languages, Japanese benefits from our tuning method the most, probably because its word order is quite different from English and the baseline may overfit to the English order. For example, in Japanese, almost all of the "case" relations are head-first and "obj" relations are modifier-first, and these patterns are exactly opposite to those in English, which serves as our source language. As a result, direct transfer models fail on most of these relations since they only learn the patterns in English. With our fine-tuning on unlabeled data, the model may get more familiar with the unusual patterns of word order and predict more correct attachment decisions (around 0.4 improvements in recalls). In Arabic and Indonesian, although not as obviously as in Japanese, the improvements are still consistent, especially on the relations of the core arguments.

4.4 When to Use Generative Models?

In unsupervised cross-lingual transfer setting, it is hard to find a system that is able to achieve state-of-the-art on all languages. As reflected by our experiments, there is a tradeoff between fitting source language and generalizing to target language – the flexibility of discriminative models results in overfitting issue and poor performance when transferred to distant languages. Unfortunately, a limited number of high-resource languages and many more low-resource languages in the world are mostly distant. This means that distant transfer is a practical challenge we face when dealing with low-resource languages. Next we try to give a preliminary guidance about which system should be used in specific transfer scenarios.

As discussed in Section 2.1, there are different types of distance metrics. Here we aim to compute the significance of correlation between the

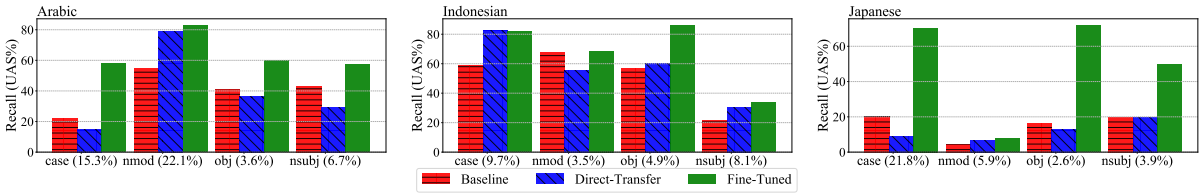

Figure 3: Results (UAS%) on typical dependency relations for Arabic, Indonesian and Japanese, respectively. "Baseline" denotes the "SelfAtt-Graph" model, and "Direct-Transfer" denotes our source model without fine-tuning. The number in the parenthesis after each dependency label indicates the relative frequency of this type.

performance difference between our method and the discriminative baseline and different distance features. We have five input distance features: geographic, genetic, syntactic, inventory, and phonological.

Specifically, we fit a generalized linear model (GLM) on the difference in accuracy and five features of all 20 target languages, then we perform a hypothesis test to compute the p-value that reflects the significance of specific features.[10] Results are shown in Table 4, where we can conclude that the genetic distance feature is significantly correlated with POS tagging performance, while geographic distance feature is significantly correlated with dependency parsing performance. As assumed before, inventory and phonological distance do not have much influence on the transfer. Interestingly, syntactic distance is not the significant term for both tasks, we posit that this is because the transfer performance is affected by both cross-lingual word embedding quality and linguistic features, thus genetic/geographic distance might be a better indicator overall. The results suggest that our method might be more suitable than the discriminative approach at genetically distant transfer for POS tagging and geographically distant transfer for parsing.

Feature	p-value	
	POS tagging	Dependency Parsing
Geographic	0.465	0.013
Genetic	0.007	0.531
Syntactic	0.716	0.231
Inventory	0.982	0.453
Phonological	0.502	0.669

Table 4: p-value of different distance features on POS tagging and dependency parsing task. A lower p-value indicates stronger association between the feature and the response, which is the difference between our method and the discriminative baselines.

4.5 Effect of Multilingual-BERT

So far the analysis and experiments of this paper focus on non-contextualized fastText word embeddings. We note that concurrently to this work, Wu and Dredze (2019) found that the recently released multilingual BERT (mBERT; Devlin et al. (2019)) is able to achieve impressive performance on various cross-lingual transfer tasks. To study the effect of contextualized mBERT word embeddings on our proposed method, we report the average POS tagging and dependency parsing results in Table 5, while detailed numbers on each language are included in Appendix C. In the mBERT experiments, all the settings and hyperparameters are the same as in Section 4.2 and Section 4.3, but the aligned fastText embeddings are replaced with the mBERT embeddings.[11] We also include the average results from fastText embeddings for comparison.

On the POS tagging task all the models greatly benefit from the mBERT embeddings, especially our method on nearby languages where the mBERT outperforms the fastText by an average of 16 absolute points. Moreover, unsupervised adaptation still considerably improves the Flow-Fix baseline, and surpasses the LSTM-CRF baseline on 9 out of 10 distant languages with an average of 6% absolute performance boost. Different from the fastText setting where our method underperforms the discriminative baseline on the nearby language group, by the use of mBERT embeddings our method also beats the discriminative baseline on 7 out of 10 nearby languages with an average of 3% absolute improvement. A major limitation of our method lies in its strong independence assumptions, which results in the failure to model the long-term context information. We posit that the contextualized word embeddings

[10] We use the GLM toolkit present in the H2O Python Module.

[11] We use the multilingual cased BERT base model released in https://github.com/google-research/bert.

emb	Tagging Disc	Tagging Flow-FT	Parsing Disc	Parsing Flow-FT
		Distant Languages		
fastText	51.93	57.10	41.73	50.02
mBERT	60.24	**66.56**	**51.86**	50.11
		Nearby Languages		
fastText	75.06	69.11	76.75	66.48
mBERT	82.17	**85.48**	**83.41**	67.70

Table 5: Average of POS tagging accuracy (%) and dependency parsing UAS (%) results, comparing mBERT and fastText. "Disc" denotes the discriminative baselines.

like mBERT exactly compensate for this drawback in our model through incorporating the context information into the observed word embeddings, so that our method is able to outperform the discriminative baseline on both distant and nearby language groups.

On dependency parsing task, however, our method does not demonstrate significant improvement by the use of mBERT, while mBERT greatly helps the discriminative baseline. Therefore, although our method still outperforms the discriminative baseline on four very distant languages, the baseline demonstrates superior performance on other languages when using mBERT. Interestingly, we find that the performance of flow-based models with mBERT is similar to the performance with fastText word embeddings. Based on this, better generative models for unsupervised dependency parsing that can take advantage of contextualized embeddings seems a promising direction for future work.

5 Related Work

Cross-lingual transfer learning has been widely studied to help induce syntactic structures in low-resource languages (McDonald et al., 2011; Täckström et al., 2013a; Agić et al., 2014; Tiedemann, 2015; Kim et al., 2017; Schuster et al., 2019; Ahmad et al., 2019). In the case when no available target annotations are available, unsupervised cross-lingual transfer can be performed by directly applying pre-trained source model to the target language. (Guo et al., 2015; Schuster et al., 2019; Ahmad et al., 2019). The challenge of direct transfer method lies in the different linguistic rules between source and distant target languages. Utilizing multiple sources of resources can mitigate this issue and has been actively studied in the past years (Cohen et al., 2011; Naseem et al., 2012; Täckström et al., 2013b; Zhang and Barzilay, 2015; Aufrant et al., 2015; Ammar et al., 2016; Wang and Eisner, 2018, 2019). Other approaches that try to overcome the lack of annotations include annotation projection by the use of bitext supervision or bilingual lexicons (Hwa et al., 2005; Smith and Eisner, 2009; Wisniewski et al., 2014) and source data point selection (Søgaard, 2011; Täckström et al., 2013b).

Learning from both labeled source data and unlabeled target data has been explored before. Cohen et al. (2011) learns a generative target language parser as a linear interpolation of multiple source language parameters, Naseem et al. (2012) and Täckström et al. (2013b) rely on additional language typological features to guide selective model parameter sharing in a multi-source transfer setting, Wang and Eisner (2018, 2019) extract linguistic features from target languages by training a feature extractor on multiple source languages.

6 Conclusion

In this work, we focus on transfer to distant languages for POS tagging and dependency parsing, and propose to learn a structured flow model in a cross-lingual setting. Through learning a new latent embedding space as well as language-specific knowledge with unlabeled target data, our method proves effective at transferring to distant languages.

Acknowledgements

This research was supported by NSF Award No. 1761548 "Discovering and Demonstrating Linguistic Features for Language Documentation."

References

Željko Agić, Jörg Tiedemann, Kaja Dobrovoljc, Simon Krek, Danijela Merkler, and Sara Može. 2014. Cross-lingual dependency parsing of related languages with rich morphosyntactic tagsets. In *EMNLP 2014 Workshop on Language Technology for Closely Related Languages and Language Variants*.

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

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

Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. 2015. Zero-resource dependency parsing: Boosting delexicalized cross-lingual transfer with linguistic knowledge. In *Proceedings of COLING*.

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

Shay B Cohen, Dipanjan Das, and Noah A Smith. 2011. Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of EMNLP*.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proceedings of ICLR*.

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

Laurent Dinh, David Krueger, and Yoshua Bengio. 2014. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of ICLR*.

Matthew S. Dryer. 2013. Order of subject, object and verb. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

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

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

Harald Hammarstrm, Robert Forkel, Martin Haspelmath, and Sebastian Bank. 2015. Glottolog 2.6. Max Planck Institute for the Science of Human History.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2018. Unsupervised learning of syntactic structure with invertible neural projections. In *Proceedings of EMNLP*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

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. 2016. Unsupervised neural dependency parsing. In *Proceedings of EMNLP*.

Joo-Kyung Kim, Young-Bum Kim, Ruhi Sarikaya, and Eric Fosler-Lussier. 2017. Cross-lingual transfer learning for pos tagging without cross-lingual resources. In *Proceedings of EMNLP*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Dan Klein and Christopher D Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of ACL*.

M. Paul Lewis, Gary F. Simons, and Charles D. Fennig, editors. 2015. *Ethnologue: Languages of the World, Eighteenth edition*. SIL International, Dallas, Texas.

Patrick Littell, David R Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. Uriel and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of EACL*.

Xuezhe Ma, Zecong Hu, Jingzhou Liu, Nanyun Peng, Graham Neubig, and Eduard Hovy. 2018. Stack-pointer networks for dependency parsing. In *Proceedings of ACL*.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of EMNLP*.

Steven Moran, Daniel McCloy, and Richard Wright, editors. 2014. *PHOIBLE Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of ACL*.

Joakim Nivre, Mitchell Abrams, Željko Agić, and et al. 2018. Universal dependencies 2.2. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*.

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

David A Smith and Jason Eisner. 2009. Parser adaptation and projection with quasi-synchronous grammar features. In *Proceedings of EMNLP*.

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *Proceedings of ICLR*.

Anders Søgaard. 2011. Data point selection for cross-language adaptation of dependency parsers. In *Proceedings of ACL*.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013a. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013b. Target language adaptation of discriminative transfer parsers. In *Proceedings of NAACL-HLT*.

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)*.

Dingquan Wang and Jason Eisner. 2018. Synthetic data made to order: The case of parsing. In *Proceedings EMNLP*.

Dingquan Wang and Jason Eisner. 2019. Surface statistics of an unknown language indicate how to parse it. *Transactions of the Association for Computational Linguistics*.

Guillaume Wisniewski, Nicolas Pécheux, Souhir Gahbiche-Braham, and François Yvon. 2014. Cross-lingual part-of-speech tagging through ambiguous learning. In *Proceedings of EMNLP*.

Shijie Wu and Mark Dredze. 2019. Beto, Bentz, Becas: The surprising cross-lingual effectiveness of BERT. *arXiv preprint arXiv:1904.09077*.

Jie Yang and Yue Zhang. 2018. NCRF++: An open-source neural sequence labeling toolkit. In *Proceedings of ACL*.

Yuan Zhang and Regina Barzilay. 2015. Hierarchical low-rank tensors for multilingual transfer parsing. In *Proceedings of EMNLP*.

Yuan Zhang, David Gaddy, Regina Barzilay, and Tommi Jaakkola. 2016. Ten pairs to tag–multilingual pos tagging via coarse mapping between embeddings. In *Proceedings of NAACL-HLT*.

A Details of UD Treebanks

Language	Dist.	Treebank		#Sent.
Chinese (zh)	0.86	GSD	train	3997
			dev	500
			test	500
Persian (fa)	0.86	Seraji	train	4798
			dev	599
			test	600
Arabic (ar)	0.86	PADT	train	6075
			dev	909
			test	680
Japanese (ja)	0.71	GSD	train	7164
			dev	511
			test	557
Indonesian (id)	0.71	GSD	train	4477
			dev	559
			test	557
Korean (ko)	0.69	GSD, Kaist	train	27410
			dev	3016
			test	3276
Turkish (tr)	0.62	IMST	train	3685
			dev	975
			test	975
Hindi (hi)	0.61	HDTB	train	13304
			dev	1659
			test	1684
Croatian (hr)	0.59	SET	train	6983
			dev	849
			test	1057
Hebrew (he)	0.57	HTB	train	5241
			dev	484
			test	491
Bulgarian (bg)	0.50	BTB	train	8907
			dev	1115
			test	1116
Italian (it)	0.50	ISDT	train	13121
			dev	564
			test	482
Portuguese (pt)	0.48	Bosque, GSD	train	17993
			dev	1770
			test	1681
French (fr)	0.46	GSD	train	14554
			dev	1478
			test	416
Spanish (es)	0.46	GSD, AnCora	train	28492
			dev	3054
			test	2147
Norwegian (no)	0.45	Bokmaal, Nynorsk	train	29870
			dev	4300
			test	3450
Danish (da)	0.41	DDT	train	4383
			dev	564
			test	565
Swedish (sv)	0.40	Talbanken	train	4303
			dev	504
			test	1219
Dutch (nl)	0.37	Alpino, LassySmall	train	18058
			dev	1394
			test	1472
German (de)	0.36	GSD	train	13814
			dev	799
			test	977
English (en)	–	EWT	train	12543
			dev	2002
			test	2077

Table 6: Statistics of the UD Treebanks that we used.

We list the statistics of the UD Treebanks that we used in the following two tables. The left one lists the distance (to English) languages and the right one lists the similar (to English) languages.

B Model Hyperparameters

We use the same architecture as in He et al. (2018) for the invertible projection function f_ϕ which is the NICE architecture (Dinh et al., 2014). It contains 8 coupling layers. The coupling function in each coupling layer is a rectified network with an input layer, one hidden layer, and linear output units. The number of hidden units is set to the same as the number of input units, which is 150 in our case. POS tagger is trained with batch size 32, while dependency parser is trained with batch size 16.

C Full Results with mBERT

Here we report in Table 7 the full results on all languages with mBERT.[12]

[12]The results of our discriminative baselines are different from the ones reported in Wu and Dredze (2019) because they do not use additional encoders on top of the pre-trained mBERT word embeddings, while we keep the models unchanged here for direct comparison with fastText embeddings. On some languages our version produces better results and sometimes their version is superior.

	POS Tagging			Dependency Parsing		
Lang	**LSTM-CRF**	**Flow-Fix**	**Flow-FT**	**SelfAtt-Graph**	**Flow-Fix**	**Flow-FT**
			Distant Languages			
zh (0.86)	59.63	53.61	**65.84**	**48.78**	35.73	35.64
fa (0.86)	57.63	56.18	**68.55**	51.47	37.99	**63.18**
ar (0.86)	53.50	48.92	**67.33**	50.91	32.13	**56.85**
ja (0.71)	**46.81**	40.98	46.06	40.08	19.23	**43.55**
id (0.71)	74.95	70.95	**78.72**	57.94	47.00	**64.35**
ko (0.69)	50.74	47.99	**54.07**	**39.42**	34.67	37.02
tr (0.62)	60.08	54.69	**61.16**	**42.80**	34.88	37.06
hi (0.61)	58.86	53.16	**68.39**	**48.44**	29.15	33.17
hr (0.59)	74.98	66.35	**78.61**	**73.63**	59.68	65.27
he (0.57)	65.24	57.27	**76.83**	**65.11**	51.39	65.03
AVG (mBERT)	60.24	55.01	**66.56**	**51.86**	38.19	50.11
AVG (fastText)	51.93	45.75	57.10	41.73	38.09	50.02
			Nearby Languages			
bg (0.50)	**82.36**	74.56	80.68	**86.32**	73.65	74.06
it (0.50)	76.70	66.02	**87.88**	**86.71**	69.09	71.59
pt (0.48)	83.45	80.83	**86.49**	**83.75**	66.67	69.56
fr (0.46)	79.22	74.21	**87.21**	**86.64**	66.08	69.14
es (0.46)	77.68	72.28	**84.50**	**81.74**	63.18	66.46
no (0.45)	**85.29**	80.69	83.96	**85.01**	65.47	66.08
da (0.41)	85.57	81.90	**86.79**	**82.22**	61.61	62.15
sv (0.41)	**86.39**	81.27	86.31	**85.33**	66.04	64.51
nl (0.40)	83.67	78.88	**85.05**	**77.32**	61.70	63.24
de (0.37)	81.37	78.97	**85.96**	**79.03**	70.19	70.19
AVG (mBERT)	82.17	76.96	**85.48**	**83.41**	66.37	67.70
AVG (fastText)	75.06	63.19	69.11	76.75	66.30	66.48
en*	95.13	91.22	–	92.84	67.76	–

Table 7: POS tagging accuracy (%) and dependency parsing UAS (%) results when using mBERT as the aligned embeddings. Numbers next to languages names are their distances to English. Supervised accuracy on English ($*$) is included for reference.