On Difficulties of Cross-Lingual Transfer with Order Differences: A Case Study on Dependency Parsing

Wasi Uddin Ahmad[1]*, **Zhisong Zhang**[2]*, **Xuezhe Ma**[2]
wasiahmad@cs.ucla.edu, {zhisongz, xuezhem}@cs.cmu.edu
Eduard Hovy[2], **Kai-Wei Chang**[1], **Nanyun Peng**[3†]
ehovy@cs.cmu.edu, kwchang@cs.ucla.edu, npeng@isi.edu

[1]University of California, Los Angeles, [2]Carnegie Mellon University
[3]University of Southern California

Abstract

Word order is a significant distinctive feature to differentiate languages (Dryer, 2007). In this paper, we investigate cross-lingual transfer and posit that an order-agnostic model will perform better when transferring to distant foreign languages. To test our hypothesis, we train dependency parsers on an English corpus and evaluate their transfer performance on 30 other languages. Specifically, we compare encoders and decoders based on Recurrent Neural Networks (RNNs) and modified self-attentive architectures. The former rely on sequential information while the latter are more flexible at modeling token order. Detailed analysis shows that RNN-based architectures transfer well to languages that are close to English, while self-attentive models have better overall cross-lingual transferability and perform especially well on distant languages.

1 Introduction

Cross-lingual learning which explores knowledge transfer between different languages has tremendous practical value. It reduces the requirement of annotated data for the target language which could be especially useful for languages that have scarce resources. It has been applied to many NLP tasks such as text categorization (Zhou et al., 2016a), tagging (Kim et al., 2017), dependency parsing (Guo et al., 2015, 2016) and machine translation (Zoph et al., 2016). On the other hand, it is a challenging problem as it requires understanding and handling of differences between languages at levels of morphology, syntax, and semantics. It is especially challenging to learn invariant features that can robustly transfer to distant languages.

Prior work on cross-lingual transfer mainly focused on word-level information by inducing multi-lingual invariant word embeddings (Xiao and Guo, 2014; Guo et al., 2016; Sil et al., 2018). However, words are not independent in sentences; their interaction and combination form larger linguistic units, known as *context*. Encoding context information is vital for most NLP problems, therefore, successful architectures for NLP usually contains mechanisms to contextualize words and compose higher-level features, such as using Convolutional Neural Networks (CNNs) or RNNs (Kim, 2014; McCann et al., 2017). We refer to these mechanisms as context encoding. In this paper, we explore transferring contextual information, where we consider how to induce language-independent context features.

For language transfer, one of the challenges is the variations of word order among different languages. For example, the Verb-Object pattern in English can hardly be found in Japanese. This challenge should be taken into consideration in model design. RNN is a prevalent family of models for many NLP tasks and has demonstrated compelling performances (Mikolov et al., 2010; Sutskever et al., 2014; Peters et al., 2018). However, its sequential nature makes it heavily reliant on word order information, which exposes to the risk of encoding language-specific order information that cannot generalize across languages. We characterize this as the "*order-sensitive*" property. Another family of models known as "Transformer" uses self-attention mechanisms to capture context information, and was shown to be effective in various NLP tasks (Vaswani et al., 2017; Liu et al., 2018; Kitaev and Klein, 2018). With modifications on position representations, the self-attention mechanism can be more flexible than RNNs at capturing context since it does not explicitly rely on word order information. We refer to this as the "*order-free*" property.

In this work, we posit that *order-free* mod-

*Equal contribution. Listed by alphabetical order.
†Corresponding author.

Language Families	Languages
Afro-Asiatic	Arabic (ar), Hebrew (he)
Austronesian	Indonesian (id)
IE.Baltic	Latvian (lv)
IE.Germanic	Danish (da), Dutch (nl), English (en), German (de), Norwegian (no), Swedish (sv)
IE.Indic	Hindi (hi)
IE.Latin	Latin (la)
IE.Romance	Catalan (ca), French (fr), Italian (it), Portuguese (pt), Romanian (ro), Spanish (es)
IE.Slavic	Bulgarian (bg), Croatian (hr), Czech (cs), Polish (pl), Russian (ru), Slovak (sk), Slovenian (sl), Ukrainian (uk)
Japanese	Japanese (ja)
Korean	Korean (ko)
Sino-Tibetan	Chinese (zh)
Uralic	Estonian (et), Finnish (fi)

Table 1: The selected languages grouped by language families. "IE" is the abbreviation of Indo-European.

els are less vulnerable to overfitting to language-specific word order features and thus have better transferability than *order-sensitive* models. To test our hypothesis, we first quantify language distance in terms of word order typology, and then systematically study the transferability of order-sensitive and order-free neural architectures on cross-lingual dependency parsing. We choose dependency parsing primarily because of the availability of unified annotations across a broad spectrum of languages (Nivre et al., 2018). Besides, word order typology is found to influence dependency parsing (Naseem et al., 2012; Täckström et al., 2013; Zhang and Barzilay, 2015; Ammar et al., 2016; Aufrant et al., 2016). Moreover, parsing is a low-level NLP task (Hashimoto et al., 2017) that can benefit many downstream applications (McClosky et al., 2011; Gamallo et al., 2012; Jie et al., 2017).

We conduct evaluations on 31 languages across a broad spectrum of language families, as shown in Table 1. Our empirical results show that *order-free* encoding and decoding models generally perform better than the *order-sensitive* ones for cross-lingual transfer, especially when the source and target languages are distant.

2 Quantifying Language Distance

Word order can be a significant distinctive feature to differentiate languages (Dryer, 2007). Since word order features can especially influence parsing, we first verify that we can measure "language

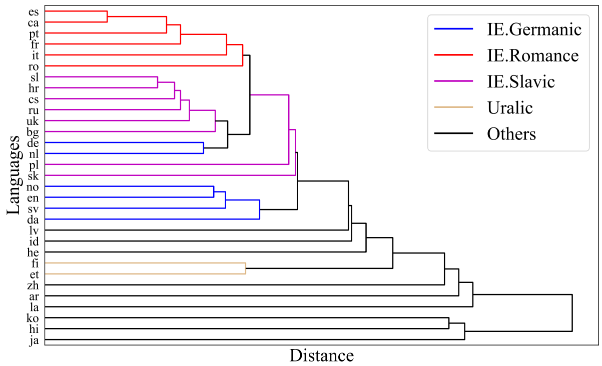


Figure 1: Hierarchical clustering (with the Nearest Point Algorithm) dendrogram of the languages by their word-ordering vectors.

distance" based on word order. The World Atlas of Language Structures (WALS) (Dryer and Haspelmath, 2013) provides a great reference for word order typology, and can be used to construct feature vectors for languages (Littell et al., 2017). But since we already have the universal dependency annotations, we take an empirical way and directly extract word order features by the directionality of different dependency relations (Liu, 2010).

We conduct our study using the Universal Dependencies (UD) Treebanks (v2.2) (Nivre et al., 2018). We select 31 languages for evaluation and analysis, with the selection criterion that the total token number in the treebanks of a language is over 100K. We group these languages by their language families in Table 1. Detailed statistical information of the selected languages and treebanks can be found in Appendix A[1].

We look at finer-grained dependency types than the 37 universal dependency labels[2] in UD v2 by augmenting the dependency labels with the universal part-of-speech (POS) tags of the head and modifier nodes. Specifically, we use triples "(ModifierPOS, HeadPOS, DependencyLabel)" as the augmented dependency types. With this, we can investigate language differences in a fine-grained way by defining directions on these triples (i.e. modifier before head or modifier after head).

We conduct feature selection by filtering out rare types as they can be unstable. This results in 52 selected types and please refer to Appendix B for more details. For each dependency type, we collect the statistics of directionality (Liu, 2010; Wang and Eisner, 2017). Since there can be only

[1]Please refer to the supplementary materials for all the appendices of this paper.

[2]http://universaldependencies.org/u/dep/index.html

two directions for an edge, for each dependency type, we use the relative frequency of the left-direction (modifier before head) as the directional feature. By concatenating the directional features of all selected augmented dependency types (triples), we obtain a word-ordering feature vector for each language. We calculate the **word-ordering distance** using these vectors. In this work, we simply use Manhattan distance, which works well as shown in our analysis (Section 4.3). We perform hierarchical clustering based on the word-ordering vectors for the selected languages, following Östling (2015). As shown in Figure 1, the grouping of the ground truth language families is almost recovered, with only two outliers German (de) and Dutch (nl), which are indeed different to English. For instance, German and Dutch adopt a larger portion of Object-Verb order in embedded clauses. The above analysis indicates that word ordering is a major feature to characterize distance between languages and should be taken as a major factor in the model designs.

3 Models

Our primary goal is to conduct cross-lingual transfer of syntactic dependencies without any annotation in the target languages. The basic structure of our experimental models is as follows. The first layer is an input embedding layer, for which we simply concatenate word and POS embeddings. The POS embeddings are trained from scratch, while the word embeddings are fixed and initialized with the multilingual embeddings by Smith et al. (2017). These inputs are fed to the encoder to get contextual representations, which is further used by the decoder for structured prediction.

For the cross-lingual transfer, we hypothesize that the models capturing less language-specific information of the source language will have better transfer ability. We focus on the word order information, and explore different encoders and decoders that are considered as *order-sensitive* and *order-free*, respectively.

3.1 Contextual Encoders

Considering the sequential nature of languages, RNN can be a natural choice for encoding. However, modeling words one by one in the sequence inevitably encodes word order information, which may be specific to the source language. To alleviate this problem, we adopt the self-attention based encoder (Vaswani et al., 2017) for cross-lingual parsing. It can be less sensitive to word order but not necessarily less potent at capturing contextual information, which makes it suitable in our setting.

RNN Encoder Following previous work (Kiperwasser and Goldberg, 2016; Dozat and Manning, 2017), we employ k-layer bidirectional LSTMs (Hochreiter and Schmidhuber, 1997) on top of the input vectors to obtain contextual representations. Since it explicitly depends on word order, we will refer it as an *order-sensitive* encoder.

Self-Attention Encoder The original self-attention encoder (Transformer) takes absolute positional embeddings as inputs, which still capture much positional and thus ordering information. To mitigate this, we utilize relative position representations (Shaw et al., 2018), with a further simple modification to make it order-agnostic: the original relative position representations discriminate left and right contexts by adding signs to distances, while we only use the distances and discard directional information. We provide more details about this modification in Appendix C. With this, the model knows only what words are surrounding but cannot tell the directions. Since self-attention encoder is much less sensitive to word order, we refer to it as an *order-free* encoder.

3.2 Structured Decoders

With the contextual representations from the encoder, the decoder predicts the output tree structures. We also investigate two types of decoders with different sensitivity to ordering information.

Stack-Pointer Decoder Recently, Ma et al. (2018) proposed a top-down transition-based decoder and obtained state-of-the-art results. Thus, we select it as our transition-based decoder. To be noted, in this Stack-Pointer decoder, RNN is utilized to record the decoding trajectory and also can be sensitive to word order. Therefore, we will refer to it as an *order-sensitive* decoder.

Graph-based Decoder Graph-based decoders assume simple factorization and can search globally for the best structure. Recently, with a deep biaffine attentional scorer, Dozat and Manning (2017) obtained state-of-the-art results with simple first-order factorization (Eisner, 1996; McDonald et al., 2005). In fact, this method resembles the self-attentive encoder in some way, and can be regarded as a self-attentive output layer. It does not depend on ordering information, and thus will be referred to as an *order-free* decoder.

4 Experiments and Analysis

In this section, we compare four architectures for cross-lingual transfer dependency parsing with different combination of order-free and order-sensitive encoder and decoder. We conduct several detailed analyses showing the pros and cons of both type of models.

4.1 Setup

Settings In our experiments, we take English as the source language and 30 other languages as target languages. We use only English for both training and hyper-parameter tuning. During testing, we directly apply the trained model to target languages with the inputs from target languages passed through pretrained multilingual embeddings that are projected into a common space as the source language. The projection is done by the offline transformation method (Smith et al., 2017) with pre-trained 300d monolingual embeddings from FastText (Bojanowski et al., 2017). We freeze word embeddings since fine-tuning on them may disturb the multi-lingual alignments.

For other hyper-parameters, we adopted similar ones as in the Biaffine Graph Parser (Dozat and Manning, 2017) and the Stack-Pointer Parser (Ma et al., 2018). Detailed hyper-parameter settings can be found in Appendix D . Throughout our experiments, we used only the first-level UD labels since fine-grained labels are language-dependent. The evaluation metrics are Unlabeled attachment score (UAS) and labeled attachment score (LAS) with punctuations excluded. We trained our cross-lingual models five times with different initializations and reported average scores.

Systems As described before, we have an *order-free* (Self-Attention) and an *order-sensitive* (BiLSTM-RNN) encoder, as well as an *order-free* (Biaffine Attention Graph-based) and an *order-sensitive* (Stack-Pointer) decoder. The combination gives us four different models, named in the format of "Encoder" plus "Decoder". For clarity, we also mark each model with their encoder-decoder order sensitivity characteristics. For example, "SelfAtt-Graph (OF-OF)" refers to the model with self-attention order-free encoder and graph-based order-free decoder. We benchmark our models with a baseline shift-reduce transition-based parser, which gave previous state-of-the-art results for single-source zero-resource cross-lingual parsing (Guo et al., 2015). Since they used older datasets, we re-trained the model on our datasets with their implementation[3]. We also list the supervised learning results using the "RNN-Graph" model on each language as a reference of the upper-line for cross-lingual parsing.

4.2 Results

The results on the test sets are shown in Table 2. The languages are ordered by their order typology distance to English. In preliminary experiments, we found our lexicalized models performed poorly on Chinese (zh) and Japanese (ja). We found the main reason was that their embeddings were not well aligned to English. Therefore, we use delexicalized models, where only POS tags are used as inputs. The delexicalized results[4] for Chinese and Japanese are listed in the rows marked with "*".

Overall, the "SelfAtt-Graph" model performs the best in over half of the languages and beats the runner-up "RNN-Graph" by around 1.3 in UAS and 1.2 in LAS on average. When compared with "RNN-Stack" and "SelfAtt-Stack", the average difference is larger than 1.5 points. This shows that models capture less word order information generally perform better at cross-lingual parsing. Compared with the baseline, our superior results show the importance of the contextual encoder. Compared with the supervised models, the cross-lingual results are still lower by a large gap, indicating space for improvements.

After taking a closer look, we find an interesting pattern in the results: RNN-based models perform better at languages that are near English (upper rows in the table), while for languages that are "distant" from English, the "SelfAtt-Graph" performs much better. Such patterns correspond well with our motivation, that is, the design of models considering word order information is crucial in cross-lingual transfer. We conduct more thorough analysis in the next subsection.

[3]https://github.com/jiangfeng1124/acl15-clnndep. We also evaluated our models on the older dataset and compared with their results, as shown in Appendix E.

[4]We found delexicalized models to be better only at zh and ja, for about 5 and 10 points respectively. For other languages, they performed worse for about 2 to 5 points. We also tried models without POS, and found them worse for about 10 points on average. We leave further investigation of input representations to future work.

Lang	Dist. to English	SelfAtt-Graph (OF-OF)	RNN-Graph (OS-OF)	SelfAtt-Stack (OF-OS)	RNN-Stack (OS-OS)	Baseline (Guo et al., 2015)	Supervised (RNN-Graph)
en	0.00	90.35/88.40	90.44/88.31	90.18/88.06	**91.82**†/**89.89**†	87.25/85.04	90.44/88.31
no	0.06	80.80/72.81	80.67/72.83	80.25/72.07	**81.75**†/**73.30**†	74.76/65.16	94.52/92.88
sv	0.07	80.98/73.17	81.23/73.49	80.56/72.77	**82.57**†/**74.25**†	71.84/63.52	89.79/86.60
fr	0.09	77.87/72.78	**78.35**†/**73.46**†	76.79/71.77	75.46/70.49	73.02/64.67	91.90/89.14
pt	0.09	**76.61**†/67.75	76.46/**67.98**	75.39/66.67	74.64/66.11	70.36/60.11	93.14/90.82
da	0.10	76.64/67.87	77.36/68.81	76.39/67.48	**78.22**†/**68.83**	71.34/61.45	87.16/84.23
es	0.12	74.49/66.44	**74.92**†/**66.91**†	73.15/65.14	73.11/64.81	68.75/59.59	93.17/90.80
it	0.12	80.80/75.82	**81.10**/**76.23**†	79.13/74.16	80.35/75.32	75.06/67.37	94.21/92.38
hr	0.13	**61.91**†/**52.86**†	60.09/50.67	60.58/51.07	60.80/51.12	52.92/42.19	89.66/83.81
ca	0.13	73.83/65.13	**74.24**†/**65.57**†	72.39/63.72	72.03/63.02	68.23/58.15	93.98/91.64
pl	0.13	**74.56**†/**62.23**†	71.89/58.59	73.46/60.49	72.09/59.75	66.74/53.40	94.96/90.68
uk	0.13	**60.05**/**52.28**†	58.49/51.14	57.43/49.66	59.67/51.85	54.10/45.26	85.98/82.21
sl	0.13	**68.21**†/**56.54**†	66.27/54.57	66.55/54.58	67.76/55.68	60.86/48.06	86.79/82.76
nl	0.14	68.55/60.26	67.88/60.11	67.88/59.46	**69.55**†/**61.55**†	63.31/53.79	90.59/87.52
bg	0.14	**79.40**†/**68.21**†	78.05/66.68	78.16/66.95	78.83/67.57	73.08/61.23	93.74/89.61
ru	0.14	60.63/51.63	59.99/50.81	59.36/50.25	**60.87**/**51.96**	55.03/45.09	94.11/92.56
de	0.14	**71.34**†/**61.62**†	69.49/59.31	69.94/60.09	69.58/59.64	65.14/54.13	88.58/83.68
he	0.14	**55.29**/**48.00**†	54.55/46.93	53.23/45.69	54.89/40.95	46.03/26.57	89.34/84.49
cs	0.14	**63.10**†/**53.80**†	61.88/52.80	61.26/51.86	62.26/52.32	56.15/44.77	94.03/91.87
ro	0.15	**65.05**†/**54.10**†	63.23/52.11	62.54/51.46	60.98/49.79	56.01/44.04	90.07/84.50
sk	0.17	**66.65**/**58.15**†	65.41/56.98	65.34/56.68	66.56/57.48	57.75/47.73	90.19/86.38
id	0.17	**49.20**†/**43.52**†	47.05/42.09	47.32/41.70	46.77/41.28	40.84/33.67	87.19/82.60
lv	0.18	70.78/49.30	**71.43**†/**49.59**	69.04/47.80	70.56/48.53	62.33/41.42	83.67/78.13
fi	0.20	66.27/48.69	**66.36**/**48.74**	64.82/47.50	66.25/48.28	58.51/38.65	88.04/85.04
et	0.20	**65.72**†/**44.87**†	65.25/44.40	64.12/43.26	64.30/43.50	56.13/34.86	86.76/83.28
zh*	0.23	**42.48**†/**25.10**†	41.53/24.32	40.56/23.32	40.92/23.45	40.03/20.97	73.62/67.67
ar	0.26	**38.12**†/**28.04**†	32.97/25.48	32.56/23.70	32.85/24.99	32.69/22.68	86.17/81.83
la	0.28	**47.96**†/**35.21**†	45.96/33.91	45.49/33.19	43.85/31.25	39.08/26.17	81.05/76.33
ko	0.33	**34.48**†/**16.40**†	33.66/15.40	32.75/15.04	33.11/14.25	31.39/12.70	85.05/80.76
hi	0.40	**35.50**†/**26.52**†	29.32/21.41	31.38/23.09	25.91/18.07	25.74/16.77	95.63/92.93
ja*	0.49	**28.18**†/**20.91**†	18.41/11.99	20.72/13.19	15.16/9.32	15.39/08.41	89.06/78.74
Average	0.17	**64.06**†/**53.82**†	62.71/52.63	62.22/52.00	62.37/51.89	57.09/45.41	89.44/85.62

Table 2: Results (UAS%/LAS%) on the test sets. Languages are sorted by the word-ordering distance to English, as shown in the second column. '*' refers to results of delexicalized models, '†' means that the best transfer model is statistically significantly better ($p < 0.05$) than all other transfer models. Models are marked with their encoder and decoder order sensitivity, OF denotes order-free and OS denotes order-sensitive.

4.3 Analysis

We further analyze how different modeling choices influence cross-lingual transfer. Since we have not touched the training sets in UD for languages other than English, to be more robust (with more data), we evaluate and analyze the results on the training sets of the target languages in this subsection (Section 4.3). Detailed results on the training sets are shown in Appendix F. The trends are similar to those on the test sets. For English, we use the results on the test set since its training and dev set is exposed in training. Because of possible issues in the bilingual word embeddings, we use delexicalized results for Chinese and Japanese.

4.3.1 On Modeling Word Order

We hypothesize that models that are less sensitive to word order can be better at cross-lingual transfer. To empirically investigate this point, we conduct controlled comparisons on various encoders with the same graph-based decoder. Table 3 shows the average performances on all languages.

To compare models with various degrees of sensitivity to word order, we include several variations of self-attention models. The "SelfAtt-NoPosi" is the self-attention model without any positional information. Although it is most insensitive to word order, it performs poorly possibly because of the lack of access to the locality of contexts. The self-attention model with absolute positional embeddings ("SelfAtt-Absolute") also does not perform well. In the case of parsing, relative positional representations may be more useful as indicated by the improvements bring by the directional relative position representations ("SelfAtt-Relative+Dir") (Shaw et al., 2018). Interestingly, the RNN encoder ranks between "SelfAtt-Relative+Dir" and "SelfAtt-Absolute"; all these

Model	UAS%	LAS%
SelfAtt-Relative (Ours)	64.57	54.14
SelfAtt-Relative+Dir	63.93	53.62
RNN	63.25	52.94
SelfAtt-Absolute	61.76	51.71
SelfAtt-NoPosi	28.18	21.45

Table 3: Comparisons of different encoders (averaged results over all languages on the original training sets).

three encoders explicitly capture word order information in some way. Finally, by discarding the information of directions, our relative position representation ("SelfAtt-Relative") performs the best (significantly better than all others at $p < 0.05$), indicating its effectiveness in capturing useful context information without depending too much on language-specific order information.

These results support our hypothesis that a model's sensitivity to word order affects its cross-lingual transfer performances. In later sections, we stick to our "SelfAtt-Relative" variation of the self attentive encoder and focus on the comparisons among the four main models.

4.3.2 On The Overall Pattern

We posit that order-free models can do better than order-sensitive ones on cross-lingual transfer parsing when the target languages have different word orders to the source language. Now we can analyze this with the word-ordering distance.

For each target language, we collect two types of distances when comparing it to English: one is the **word-ordering distance** as described in Section 2, the other is the **performance distance**, which is the gap of evaluation scores[5] between the target language and English. The performance distance can represent the general transferability from English to this language. We calculate the correlation of these two distances on all the concerned languages, and the results turn to be quite high: the Pearson and Spearman correlations are **around 0.90 and 0.87** respectively, using the evaluations of any of our four cross-lingual transfer models. This suggests that word order is indeed an essential factor of cross-lingual transferability.

Furthermore, we individually analyze the encoders and decoders of the dependency parsers. Since we have two architectures for each of the modules, when examining one, we take the highest scores obtained by any of the other module. For example, when comparing RNN and Self-Attention encoders, we take the best evaluation scores of "RNN-Graph" and "RNN-Stack" for RNN and the best of "SelfAtt-Graph" and "SelfAtt-Stack" for Self-Attention. Figure 2 shows the score differences of encoding and decoding architectures against the languages' distances to English. For both the encoding and decoding module, we observe a similar overall pattern: the order-free models in general perform better than order-sensitive ones in the languages that are distant from the source language English. On the other hand, for some languages that are closer to English, order-sensitive models perform better, possibly benefiting from being able to capture similar word ordering information. The performance gap of order-free and order-sensitive models are positively correlated with language distance.

[5]In the rest of this paper, we simply average UAS and LAS for evaluation scores unless otherwise noted.

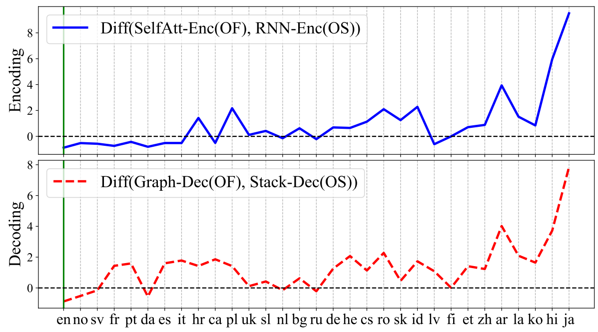
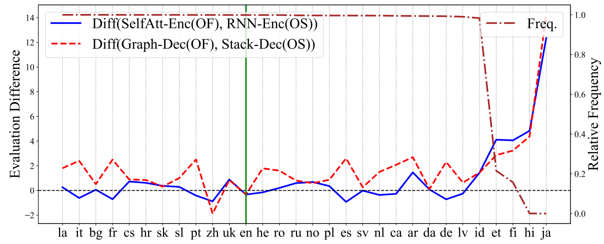


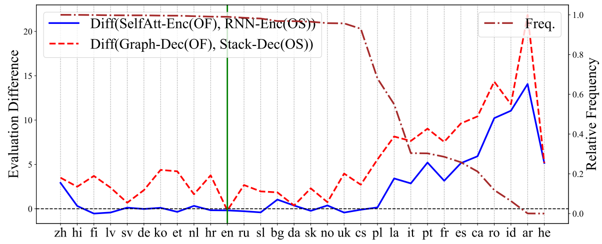
Figure 2: Evaluation score differences between Order-Free (OF) and Order Sensitive (OS) modules. We show results of both encoder (blue solid curve) and decoder (dashed red curve). Languages are sorted by their word-ordering distances to English from left to right. The position of English is marked with a green bar.

4.3.3 On Dependency Types

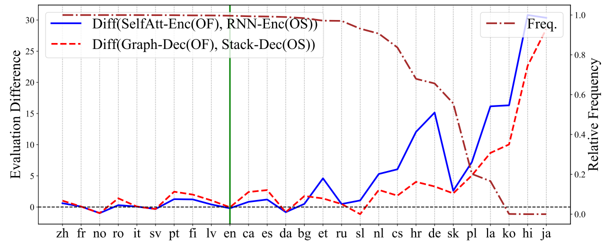
Moreover, we compare the results on specific dependency types using concrete examples. For each type, we sort the languages by their relative frequencies of left-direction (modifier before head) and plot the performance differences for encoders and decoders. We highlight the source language English in green. Figure 3 shows four typical example types: Adposition and Noun, Adjective and Noun, Auxiliary and Verb, and Object and Verb. In Figure 3a, we examine the "case" dependency type between adpositions and nouns. The pattern is similar to the overall pattern. For languages that mainly use prepositions as in English, different models perform similarly, while for languages that use postpositions, order-free models get better



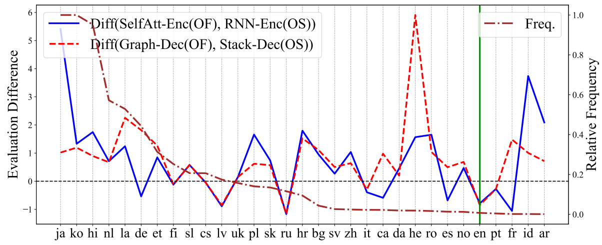
(a) Adposition & Noun (ADP, NOUN, case)



(b) Adjective & Noun (ADJ, NOUN, amod)



(c) Auxiliary & Verb (AUX, VERB, aux)



(d) Object & Verb (NOUN, VERB, obj)

Figure 3: Analysis on specific dependency types. To save space, we merge the curves of encoders and decoders into one figure. The blue and red curves and left y-axis represent the differences in evaluation scores, the brown curve and right y-axis represents the relative frequency of left-direction (modifier before head) on this type. The languages (x-axis) are sorted by this relative frequency from high to low.

results. The patterns of adjective modifier (Figure 3b) and auxiliary (Figure 3c) are also similar.

On dependencies between verbs and object nouns, although in general order-free models perform better, the pattern diverges from what we expect. There can be several possible explanations for this. Firstly, the tokens which are noun objects of verbs only take about 3.1% on average over all tokens. Considering just this specific dependency type, the correlation between frequency distances and performance differences is 0.64, which is far less than 0.9 when considering all types. Therefore, although Verb-Object ordering is a typical example, we cannot take it as the whole story of word order. Secondly, Verb-Object dependencies can often be difficult to decide. They sometimes are long-ranged and have complex interactions with other words. Therefore, merely reducing modeling order information can have complicated effects. Moreover, although our relative-position self-attention encoder does not explicitly encode word positions, it may still capture some positional information with relative distances. For example, the words in the middle of a sentence will have different distance patterns from those at the beginning or the end. With this knowledge, the model can still prefer the pattern where a verb is in the middle as in English's Subject-Verb-Object ordering and may find sentences in Subject-Object-Verb languages strange. It will be interesting to explore more ways to weaken or remove this bias.

4.3.4 On Dependency Distances

We now look into dependency lengths and directions. Here, we combine dependency length and direction into dependency distance d, by using negative signs for dependencies with left-direction (modifier before head) and positive for right-direction (head before modifier). We find a seemingly strange pattern at dependency distances $|d|$=1: for all transfer models, evaluation scores on d=-1 can reach about 80, but on d=1, the scores are only around 40. This may be explained by the relative frequencies of dependency distances as shown in Table 4, where there is a discrepancy between English and the average of other languages at d=1. About 80% of the dependencies with $|d|$=1 in English is the left direction (modifier before head), while overall other languages have more right directions at $|d|$=1. This suggests an interesting future direction of training on more source languages with different dependency distance distributions.

We further compare the four models on the d=1 dependencies and as shown in Figure 4, the familiar pattern appears again. The order-free models perform better at the languages which have more d=1 dependencies. Such finding indicates that our model design of reducing the ability to capture word order information can help on short-

d	English	Average
<-2	14.36	12.93
-2	15.45	11.83
-1	31.55	30.42
1	7.51	14.22
2	9.84	10.49
>2	21.29	20.11

Table 4: Relative frequencies (%) of dependency distances. English differs from the Average at d=1.

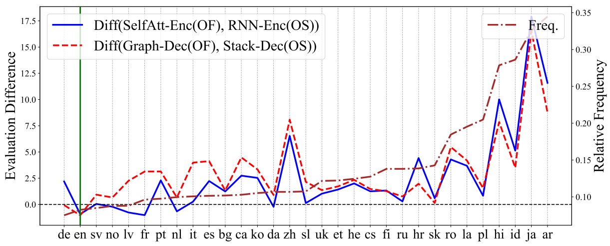


Figure 4: Evaluation differences of models on d=1 dependencies. Annotations are the same as in Figure 3, languages are sorted by percentages (represented by the brown curve and right y-axis) of d=1 dependencies.

ranged dependencies of different directions to the source language. However, the improvements are still limited. One of the most challenging parts of unsupervised cross-lingual parsing is modeling cross-lingually shareable and language-unspecific information. In other words, we want flexible yet powerful models. Our exploration of the order-free self-attentive models is a first step.

5 Related Work

Cross-language transfer learning employing deep neural networks has widely been studied in the areas of natural language processing (Ma and Xia, 2014; Guo et al., 2015; Kim et al., 2017; Kann et al., 2017; Cotterell and Duh, 2017), speech recognition (Xu et al., 2014; Huang et al., 2013), and information retrieval (Vulić and Moens, 2015; Sasaki et al., 2018; Litschko et al., 2018). Learning the language structure (e.g., morphology, syntax) and transferring knowledge from the source language to the target language is the main underneath challenge, and has been thoroughly investigated for a wide variety of NLP applications, including sequence tagging (Yang et al., 2016; Buys and Botha, 2016), name entity recognition (Xie et al., 2018), dependency parsing (Tiedemann, 2015; Agić et al., 2014), entity coreference resolution and linking (Kundu et al., 2018; Sil et al., 2018), sentiment classification (Zhou et al., 2015, 2016b), and question answering (Joty et al., 2017).

Existing work on unsupervised cross-lingual dependency parsing, in general, trains a dependency parser on the source language and then directly run on the target languages. Training of the monolingual parsers are often delexicalized, i.e., removing all lexical features from the source treebank (Zeman and Resnik, 2008; McDonald et al., 2013b), and the underlying feature model is selected from a shared part-of-speech (POS) representation utilizing the Universal POS Tagset (Petrov et al., 2012). Another pool of prior work improves the delexicalized approaches by adapting the model to fit the target languages better. Cross-lingual approaches that facilitate the usage of lexical features includes choosing the source language data points suitable for the target language (Søgaard, 2011; Täckström et al., 2013), transferring from multiple sources (McDonald et al., 2011; Guo et al., 2016; Täckström et al., 2013), using cross-lingual word clusters (Täckström et al., 2012) and lexicon mapping (Xiao and Guo, 2014; Guo et al., 2015). In this paper, we consider single-source transfer–train a parser on a single source language, and evaluate it on the target languages to test the transferability of neural architectures.

Multilingual transfer (Ammar et al., 2016; Naseem et al., 2012; Zhang and Barzilay, 2015) is another broad category of techniques applied to parsing where knowledge from many languages having a common linguistic typology are utilized. Recent works (Aufrant et al., 2016; Wang and Eisner, 2018a,b) demonstrated the significance of explicitly extracting and modeling linguistic properties of the target languages to improve cross-lingual dependency parsing. Our work is different in that we focus on the neural architectures and explore their influences on cross-lingual transfer.

6 Conclusion

In this work, we conduct a comprehensive study on how the design of neural architectures affects cross-lingual transfer learning. We examine two notable families of neural architectures (sequential RNN v.s. self-attention) using dependency parsing as the evaluation task. We show that *order-free* models perform better than *order-sensitive* ones when there is a large difference in the word order typology between the target and source language.

In future, we plan to explore multi-source transfer and incorporating prior linguistic knowledge into the models for better cross-lingual transfer.

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*.

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

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

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

Jan Buys and Jan A. Botha. 2016. Cross-lingual morphological tagging for low-resource languages. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1954–1964.

Ryan Cotterell and Kevin Duh. 2017. Low-resource named entity recognition with cross-lingual, character-level neural conditional random fields. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. volume 2, pages 91–96.

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

Matthew S Dryer. 2007. Word order. *Language typology and syntactic description* 1:61–131.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Jason M Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, pages 340–345.

Pablo Gamallo, Marcos Garcia, and Santiago Fernández-Lanza. 2012. Dependency-based open information extraction. In *Proceedings of the joint workshop on unsupervised and semi-supervised learning in NLP*. Association for Computational Linguistics, pages 10–18.

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)*. volume 1, pages 1234–1244.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2016. A representation learning framework for multi-source transfer parsing. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI'16, pages 2734–2740.

Kazuma Hashimoto, caiming xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple nlp tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1923–1933.

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

Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong. 2013. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, pages 7304–7308.

Zhanming Jie, Aldrian Obaja Muis, and Wei Lu. 2017. Efficient dependency-guided named entity recognition. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. pages 3457–3465.

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

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2017. One-shot neural cross-lingual transfer for paradigm completion. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* page 19932003.

Joo-Kyung Kim, Young-Bum Kim, Ruhi Sarikaya, and Eric Fosler-Lussier. 2017. Cross-lingual transfer learning for pos tagging without cross-lingual resources. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 2832–2838.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1746–1751.

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

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 2676–2686.

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

Robert Litschko, Goran Glavaš, Simone Paolo Ponzetto, and Ivan Vulić. 2018. Unsupervised cross-lingual information retrieval using monolingual data only. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. SIGIR '18, pages 1253–1256.

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 the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, Valencia, Spain, pages 8–14.

Haitao Liu. 2010. Dependency direction as a means of word-order typology: A method based on dependency treebanks. *Lingua* 120(6):1567–1578.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *Internation Conference on Learning Representations* .

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

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

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*. pages 6294–6305.

David McClosky, Mihai Surdeanu, and Christopher D Manning. 2011. Event extraction as dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pages 1626–1635.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL-2005*. Ann Arbor, Michigan, USA, pages 91–98.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013a. Universal dependency annotation for multilingual parsing. In *Proceedings of ACL-2013*. Sofia, Bulgaria, pages 92–97.

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

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

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

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

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.

Robert Östling. 2015. Word order typology through multilingual word alignment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. volume 2, pages 205–211.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference* *of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, pages 2227–2237.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of LREC-2012*. Istanbul, Turkey, pages 2089–2096.

Shota Sasaki, Shuo Sun, Shigehiko Schamoni, Kevin Duh, and Kentaro Inui. 2018. Cross-lingual learning-to-rank with shared representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. volume 2, pages 458–463.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, pages 464–468.

Avirup Sil, Gourab Kundu, Radu Florian, and Wael Hamza. 2018. Neural cross-lingual entity linking. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. pages 5464–5472.

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

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

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.

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

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 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 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*. Association for Computational Linguistics, pages 477–487.

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.

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

Ivan Vulić and Marie-Francine Moens. 2015. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. ACM, pages 363–372.

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

Dingquan Wang and Jason Eisner. 2018a. Surface statistics of an unknown language indicate how to parse it. *Transactions of the Association for Computational Linguistics (TACL)* .

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

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.

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

Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee. 2014. Cross-language transfer learning for deep neural network based speech enhancement. In *Chinese Spoken Language Processing (ISCSLP), 2014 9th International Symposium on*. IEEE, pages 336–340.

Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. 2016. Multi-task cross-lingual sequence tagging from scratch. *arXiv preprint arXiv:1603.06270* .

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 Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1857–1867.

Huiwei Zhou, Long Chen, Fulin Shi, and Degen Huang. 2015. Learning bilingual sentiment word embeddings for cross-language sentiment classification. 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)*. volume 1, pages 430–440.

Joey Tianyi Zhou, Sinno Jialin Pan, Ivor W. Tsang, and Shen-Shyang Ho. 2016a. Transfer learning for cross-language text categorization through active correspondences construction. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI'16, pages 2400–2406.

Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016b. Cross-lingual sentiment classification with bilingual document representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 1403–1412.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1568–1575.

Supplementary Material: Appendices

A Details of UD Treebanks

The statistics of the Universal Dependency treebanks we used are summarized in Table 5.

Language	Lang. Family	Treebank		#Sent.	#Token(w/o punct)
Arabic (ar)	Afro-Asiatic	PADT	train	6075	223881(206041)
			dev	909	30239(27339)
			test	680	28264(26171)
Bulgarian (bg)	IE.Slavic	BTB	train	8907	124336(106813)
			dev	1115	16089(13822)
			test	1116	15724(13456)
Catalan (ca)	IE.Romance	AnCora	train	13123	417587(371981)
			dev	1709	56482(50452)
			test	1846	57902(51459)
Chinese (zh)	Sino-Tibetan	GSD	train	3997	98608(84988)
			dev	500	12663(10890)
			test	500	12012(10321)
Croatian (hr)	IE.Slavic	SET	train	6983	154055(135206)
			dev	849	19543(17211)
			test	1057	23446(20622)
Czech (cs)	IE.Slavic	PDT,CAC, CLTT,FicTree	train	102993	1806230(1542805)
			dev	11311	191679(163387)
			test	12203	205597(174771)
Danish (da)	IE.Germanic	DDT	train	4383	80378(69219)
			dev	564	10332(8951)
			test	565	10023(8573)
Dutch (nl)	IE.Germanic	Alpino, LassySmall	train	18058	261180(228902)
			dev	1394	22938(19645)
			test	1472	22622(19734)
English (en)	IE.Germanic	EWT	train	12543	204585(180303)
			dev	2002	25148(21995)
			test	2077	25096(21898)
Estonian (et)	Uralic	EDT	train	20827	287859(240496)
			dev	2633	37219(30937)
			test	2737	41273(34837)
Finnish (fi)	Uralic	TDT	train	12217	162621(138324)
			dev	1364	18290(15631)
			test	1555	21041(17908)
French (fr)	IE.Romance	GSD	train	14554	356638(316780)
			dev	1478	35768(31896)
			test	416	10020(8795)
German (de)	IE.Germanic	GSD	train	13814	263804(229338)
			dev	799	12486(10809)
			test	977	16498(14132)
Hebrew (he)	Afro-Asiatic	HTB	train	5241	137680(122122)
			dev	484	11408(10050)
			test	491	12281(10895)
Hindi (hi)	IE.Indic	HDTB	train	13304	281057(262389)
			dev	1659	35217(32850)
			test	1684	35430(33010)
Indonesian (id)	Austronesian	GSD	train	4477	97531(82617)
			dev	559	12612(10634)
			test	557	11780(10026)
Italian (it)	IE.Romance	ISDT	train	13121	276019(244632)
			dev	564	11908(10490)
			test	482	10417(9237)
Japanese (ja)	Japanese	GSD	train	7164	161900(144045)
			dev	511	11556(10326)
			test	557	12615(11258)
Korean (ko)	Korean	GSD, Kaist	train	27410	353133(312481)
			dev	3016	37236(32770)
			test	3276	40043(35286)
Latin (la)	IE.Latin	PROIEL	train	15906	171928(171928)
			dev	1234	13939(13939)
			test	1260	14091(14091)
Latvian (lv)	IE.Baltic	LVTB	train	5424	80666(66270)
			dev	1051	14585(11487)

			test	1228	15073(11846)
Norwegian (no)	IE.Germanic	Bokmaal, Nynorsk	train dev test	29870 4300 3450	489217(432597) 67619(59784) 54739(48588)
Polish (pl)	IE.Slavic	LFG, SZ	train dev test	19874 2772 2827	167251(136504) 23367(19144) 23920(19590)
Portuguese (pt)	IE.Romance	Bosque, GSD	train dev test	17993 1770 1681	462494(400343) 42980(37244) 41697(36100)
Romanian (ro)	IE.Romance	RRT	train dev test	8043 752 729	185113(161429) 17074(14851) 16324(14241)
Russian (ru)	IE.Slavic	SynTagRus	train dev test	48814 6584 6491	870474(711647) 118487(95740) 117329(95799)
Slovak (sk)	IE.Slavic	SNK	train dev test	8483 1060 1061	80575(65042) 12440(10641) 13028(11208)
Slovenian (sl)	IE.Slavic	SSJ, SST	train dev test	8556 734 1898	132003(116730) 14063(12271) 24092(22017)
Spanish (es)	IE.Romance	GSD, AnCora	train dev test	28492 3054 2147	827053(730062) 89487(78951) 64617(56973)
Swedish (sv)	IE.Germanic	Talbanken	train dev test	4303 504 1219	66645(59268) 9797(8825) 20377(18272)
Ukrainian (uk)	IE.Slavic	IU	train dev test	4513 577 783	75098(60976) 10371(8381) 14939(12246)

Table 5: Statistics of the UD Treebanks we used. For language family, "IE" is the abbreviation for Indo-European. "(w/o) punct" means the numbers of the tokens excluding "PUNCT" and "SYM".

B Details about augmented dependency types

Type	Avg. Freq. (%)	#Lang.	Type	Avg. Freq. (%)	#Lang.
(ADP, NOUN, case)	7.47	31	(PROPN, VERB, nsubj)	0.81	30
(PUNCT, VERB, punct)	6.91	30	(PRON, VERB, obj)	0.77	30
(NOUN, NOUN, nmod)	4.97	31	(NOUN, ROOT, root)	0.66	31
(ADJ, NOUN, amod)	4.92	31	(VERB, VERB, xcomp)	0.61	28
(DET, NOUN, det)	4.69	30	(VERB, VERB, ccomp)	0.60	30
(VERB, ROOT, root)	4.31	31	(ADP, PRON, case)	0.57	29
(NOUN, VERB, obl)	3.96	30	(AUX, NOUN, cop)	0.57	28
(NOUN, VERB, obj)	3.10	31	(ADV, ADJ, advmod)	0.54	29
(NOUN, VERB, nsubj)	2.89	31	(AUX, ADJ, cop)	0.50	27
(PUNCT, NOUN, punct)	2.75	30	(PROPN, VERB, obl)	0.48	29
(ADV, VERB, advmod)	2.43	31	(PRON, VERB, obl)	0.44	30
(AUX, VERB, aux)	2.29	28	(ADV, NOUN, advmod)	0.41	28
(PRON, VERB, nsubj)	1.53	30	(ADJ, ROOT, root)	0.39	29
(ADP, PROPN, case)	1.46	29	(PRON, NOUN, nmod)	0.39	22
(NOUN, NOUN, conj)	1.32	30	(NOUN, ADJ, obl)	0.37	25
(VERB, NOUN, acl)	1.31	31	(PROPN, PROPN, conj)	0.35	29
(SCONJ, VERB, mark)	1.27	28	(NOUN, ADJ, nsubj)	0.35	30
(CCONJ, VERB, cc)	1.18	30	(CCONJ, ADJ, cc)	0.29	28
(PROPN, NOUN, nmod)	1.14	30	(PUNCT, NUM, punct)	0.26	24
(CCONJ, NOUN, cc)	1.13	30	(NOUN, NOUN, nsubj)	0.25	31
(NUM, NOUN, nummod)	1.11	31	(ADJ, ADJ, conj)	0.25	26
(PROPN, PROPN, flat)	1.09	26	(CCONJ, PROPN, cc)	0.22	26
(VERB, VERB, conj)	1.05	30	(PRON, VERB, iobj)	0.21	21
(PUNCT, PROPN, punct)	0.94	29	(ADV, ADV, advmod)	0.19	21
(VERB, VERB, advcl)	0.89	30	(NOUN, NOUN, appos)	0.18	23
(PUNCT, ADJ, punct)	0.89	30	(PROPN, VERB, obj)	0.17	24

Table 6: Selected augmented dependency types sorted by their average frequencies. "#Lang." denotes in how many languages the specific type appears. Our selecting criterion is "$Freq > 0.1\%$ and $\#Lang \geq 20$".

C Relative Positional Self-Attention Encoder

In this section, we briefly describe the relative position mechanism used in our self-attention encoder. Generally, it is similar to the one in (Shaw et al., 2018), with a simple modification of discarding directional information.

We directly base our descriptions on those in (Shaw et al., 2018). For the relative positional self-attention encoder, each layer calculates multiple attention heads. In each head, the input sequences $\mathbf{x} = (x_1, \ldots, x_n)$ are transformed into the output sequences $\mathbf{z} = (z_1, \ldots, z_n)$, based on the self-attention mechanism:

$$z_i = \sum_{j=1}^{n} \alpha_{ij}(x_j \cdot W^V + a_{ij}^V)$$
$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{n} \exp e_{ik}}$$
$$e_{ik} = \frac{x_i \cdot W^Q(x_j \cdot W^K + a_{ij}^K)^T}{\sqrt{d_z}}$$

Here, a_{ij}^V and a_{ij}^K are relative positional representations for the two position i and j. Similarly, we clip the distance with a maximum threshold k (which is empirically set to 10), but we do not discriminate positive and negative values. Instead, since we do not want the model to be aware of directional information, we use the absolute values of the position differences:

$$a_{ij}^K = w_{clip(|j-i|,k)}^K$$
$$a_{ij}^V = w_{clip(|j-i|,k)}^V$$
$$clip(x, k) = min(k, |x|)$$

Therefore, the learnable relative postion representations have $k + 1$ labels rather than $2k + 1$: we have $w^K = (w_0^K, \ldots, w_k^K)$, and $w^V = (w_0^V, \ldots, w_k^V)$. In this way, for one word, the model only knows the relative distances of other words, but is not explicitly told the directions of the contextual words.

D Hyper-Parameters

Table 7 summarizes the hyper-parameters that we used in our experiments. Most of them are similar to those in (Dozat and Manning, 2017) and (Ma et al., 2018).

<table>
<tr><th></th><th>Layer</th><th>Hyper-Parameter</th><th>Value</th></tr>
<tr><td rowspan="2">Input</td><td>Word</td><td>dimension</td><td>300</td></tr>
<tr><td>POS</td><td>dimension</td><td>50</td></tr>
<tr><td rowspan="8">RNN</td><td rowspan="2">Encoder</td><td>encoder layer</td><td>3</td></tr>
<tr><td>encoder size</td><td>300</td></tr>
<tr><td rowspan="2">MLP</td><td>arc MLP size</td><td>512</td></tr>
<tr><td>label MLP size</td><td>128</td></tr>
<tr><td rowspan="4">Training</td><td>Dropout</td><td>0.33</td></tr>
<tr><td>optimizer</td><td>Adam</td></tr>
<tr><td>learning rate</td><td>0.001</td></tr>
<tr><td>batch size</td><td>32</td></tr>
<tr><td rowspan="9">Self-Attention</td><td rowspan="3">Encoder</td><td>encoder layer</td><td>6</td></tr>
<tr><td>d_{model}</td><td>350</td></tr>
<tr><td>d_{ff}</td><td>512</td></tr>
<tr><td rowspan="2">MLP</td><td>arc MLP size</td><td>512</td></tr>
<tr><td>label MLP size</td><td>128</td></tr>
<tr><td rowspan="4">Training</td><td>Dropout</td><td>0.2</td></tr>
<tr><td>optimizer</td><td>Adam</td></tr>
<tr><td>learning rate</td><td>0.0001</td></tr>
<tr><td>batch size</td><td>80</td></tr>
</table>

Table 7: Hyper-parameters in our experiments.

E Results on Google Universal Dependency Treebanks v2.0

We also ran our models on Google Universal Dependency Treebanks v2.0 (McDonald et al., 2013a), which is an older dataset that was used by (Guo et al., 2015). The results show that our models perform better consistently.

Language	SelfAtt-Graph	RNN-Graph	SelfAtt-Stack	RNN-Stack	(Guo et al., 2015)
German	65.03/55.03	64.60/54.57	63.63/54.40	**65.51/55.82**	60.35/51.54
French	74.45/63.28	**76.75/65.20**	73.63/62.76	75.13/64.44	72.93/63.12
Spanish	72.00/61.50	73.99/63.46	71.73/61.42	**74.13/64.00**	71.90/62.28

Table 8: Comparisons (UAS%/LAS%) on Google Universal Dependency Treebanks v2.0.

F Results on the original training sets

Language	SelfAtt-Graph	RNN-Graph	SelfAtt-Stack	RNN-Stack
en°	90.35/88.40	90.44/88.31	90.18/88.06	**91.82/89.89**
no	80.72/72.45	80.59/72.41	80.06/71.60	**81.46/72.75**
sv	80.07/71.91	80.42/**72.39**	79.45/71.28	**80.87**/72.25
fr	79.31/74.73	**79.99/75.52**	78.62/74.02	76.84/72.22
pt	77.06/69.33	**77.33/69.91**	75.84/68.22	75.39/67.75
da	75.75/67.12	75.95/67.41	75.18/66.55	**76.98/67.50**
es	73.91/66.48	**74.39/67.03**	72.84/65.38	72.46/64.78
it	80.37/75.48	**80.89/75.99**	79.15/74.17	79.05/73.91
hr	**61.57/52.40**	59.74/50.37	59.94/50.43	60.44/50.68
ca	74.40/65.73	**74.94/66.21**	73.01/64.42	72.75/63.68
pl	**75.32/63.26**	73.12/59.76	74.28/61.46	73.21/61.02
uk	65.70/**57.48**	64.77/56.40	64.10/55.83	**65.82**/57.13
sl	**69.13/58.92**	67.35/56.87	67.74/57.08	68.95/58.26
nl	68.98/60.00	68.37/59.52	68.22/59.02	**69.16/60.11**
bg	**80.25/68.88**	78.39/67.03	79.19/67.66	79.66/68.22
ru	60.50/51.35	59.55/50.17	59.01/49.71	**60.71/51.57**
de	**67.23/58.27**	66.64/57.48	66.10/56.89	65.88/56.63
he	58.32/**49.80**	57.75/49.07	56.36/47.62	**58.79**/43.83
cs	**63.04/53.92**	61.75/52.91	61.11/51.91	62.21/52.48
ro	**65.31/54.22**	63.17/52.16	63.03/51.95	61.78/50.52
sk	**76.07/62.75**	74.67/61.15	75.93/61.97	75.37/60.94
id	**47.92/41.93**	45.07/39.91	46.23/40.16	45.62/39.67
lv	71.69/50.43	**72.48/50.85**	70.24/48.97	71.60/49.56
fi	64.64/46.21	64.63/**46.22**	63.07/44.82	**64.74**/46.09
et	**66.63/45.58**	65.78/45.01	64.94/44.04	65.06/44.33
zh*	**41.05/23.85**	40.11/23.02	39.49/22.68	39.89/22.49
ar	**38.74/28.24**	33.66/25.44	34.25/24.69	33.31/24.86
la	**49.04/35.48**	47.12/34.36	46.78/33.56	45.26/31.97
ko	**34.62/15.14**	33.91/14.16	32.70/13.77	32.95/13.14
hi	**36.01/27.24**	29.59/21.75	32.02/23.79	26.37/18.56
ja*	**28.19/21.74**	18.23/12.68	20.53/13.78	15.21/10.37
Average	**64.57**/54.14	63.25/52.94	62.88/52.44	62.88/52.16

Table 9: Results (average UAS%/LAS% over 5 runs) on the original training sets. (Languages are sorted by the word-ordering distance to English, '*' refers to results of delexicalized models, 'en°' means that for English we use results on the test set since models are trained with the English training set.)

G Results on specific dependency types for Czech

In table 10, we show results of Czech on some dependency types with evaluation breakdowns on dependency directions. We select Czech mainly for two reasons: (1) It has the largest dataset; (2) Czech is famous for relatively flexible word order. Generally, we can see that models that are more flexible on word ordering perform better. Interestingly, for objective and subjective types, we can see that LAS scores for all models are quite low even when the correct heads are predicted. The reason might be that even the relative-positional self-attention encoder can capture some positional information which further reveals word ordering information in some way.

(ADP, NOUN, case): (mod-first% in English is 99.92%.)					
Direction	Percentage	SelfAtt-Graph	RNN-Graph	SelfAtt-Stack	RNN-Stack
mod-first	99.99%	**75.34/75.34**	74.62/74.61	74.46/74.43	74.17/74.08
head-first	0.01%	–	–	–	–
all	100.00%	**75.33/75.33**	74.61/74.61	74.45/74.43	74.17/74.07

(NOUN, NOUN, nmod): (mod-first% in English is 4.72%.)					
Direction	Percentage	SelfAtt-Graph	RNN-Graph	SelfAtt-Stack	RNN-Stack
mod-first	0.97%	–	–	–	–
head-first	99.03%	21.38/17.85	18.55/16.20	20.49/16.61	**22.51/19.16**
all	100.00%	21.64/17.68	18.86/16.05	20.77/16.45	**22.78/18.98**

(ADJ, NOUN, amod): (mod-first% in English is 99.01%.)					
Direction	Percentage	SelfAtt-Graph	RNN-Graph	SelfAtt-Stack	RNN-Stack
mod-first	92.99%	88.93/88.92	**89.42/89.41**	85.39/85.21	87.26/86.37
head-first	7.01%	**41.80/37.03**	36.52/32.36	34.82/27.19	40.59/19.85
all	100.00%	85.63/85.29	**85.72/85.41**	81.85/81.14	83.98/81.71

(NOUN, VERB, obl): (mod-first% in English is 9.62%.)					
Direction	Percentage	SelfAtt-Graph	RNN-Graph	SelfAtt-Stack	RNN-Stack
mod-first	37.80%	48.84/40.33	46.39/38.49	48.75/41.08	**50.16/41.64**
head-first	62.20%	**62.81/55.97**	60.38/53.41	62.22/55.37	61.73/55.32
all	100.00%	**57.53**/50.06	55.09/47.77	57.13/49.97	57.36/**50.15**

(NOUN, VERB, obj): (mod-first% in English is 0.72%.)					
Direction	Percentage	SelfAtt-Graph	RNN-Graph	SelfAtt-Stack	RNN-Stack
mod-first	20.65%	55.56/**0.64**	53.75/0.46	54.08/0.37	**60.34**/0.18
head-first	79.35%	**73.18/65.24**	71.30/62.28	72.12/63.81	72.76/64.65
all	100.00%	69.54/**51.90**	67.68/49.52	68.39/50.71	**70.20**/51.34

(NOUN, VERB, nsubj): (mod-first% in English is 85.07%.)					
Direction	Percentage	SelfAtt-Graph	RNN-Graph	SelfAtt-Stack	RNN-Stack
mod-first	60.22%	**61.42**/58.33	58.12/54.51	60.88/58.24	60.67/**58.98**
head-first	39.78%	**64.07**/3.83	62.93/3.18	62.38/2.97	59.94/**4.42**
all	100.00%	**62.47**/36.65	60.03/34.09	61.48/36.25	60.38/**37.28**

(ADV, VERB, advmod): (mod-first% in English is 58.82%.)					
Direction	Percentage	SelfAtt-Graph	RNN-Graph	SelfAtt-Stack	RNN-Stack
mod-first	70.15%	**88.23/87.49**	86.43/85.48	86.65/85.30	86.64/83.72
head-first	29.85%	**65.79/65.28**	65.02/64.33	65.33/64.35	61.93/60.53
all	100.00%	**81.53/80.86**	80.04/79.17	80.29/79.05	79.26/76.80

(AUX, VERB, aux): (mod-first% in English is 99.64%.)					
Direction	Percentage	SelfAtt-Graph	RNN-Graph	SelfAtt-Stack	RNN-Stack
mod-first	83.71%	88.78/**88.19**	84.44/83.52	**89.03**/86.59	82.54/76.33
head-first	16.29%	**68.18/65.28**	54.59/50.87	63.96/54.02	56.67/20.24
all	100.00%	**85.42/84.46**	79.57/78.20	84.94/81.28	78.32/67.19

(VERB, VERB, advcl): (mod-first% in English is 31.02%.)					
Direction	Percentage	SelfAtt-Graph	RNN-Graph	SelfAtt-Stack	RNN-Stack
mod-first	41.75%	57.51/**55.61**	56.98/55.60	**57.54**/55.03	54.74/51.66
head-first	58.25%	**71.52/56.68**	67.39/56.08	67.27/54.17	65.93/54.13
all	100.00%	**65.67/56.23**	63.04/55.88	63.21/54.53	61.26/53.10

Table 10: Evaluation breakdowns (UAS%/LAS%) on dependency directions for Czech on some specific dependency types. "mod-first" means the dependency edges whose modifier is before head, "head-first" means the opposite, and "all" indicates both "mod-first" and "head-first". "–" replaces results that are unstable because of rare appearance (below 1%).