A systematic comparison of methods for low-resource dependency parsing on genuinely low-resource languages

Clara Vania[1]   **Yova Kementchedjhieva[2]**   **Anders Søgaard[2]**   **Adam Lopez[1]**

[1]School of Informatics, University of Edinburgh, UK

[2]University of Copenhagen, Copenhagen, Denmark

`c.vania@ed.ac.uk, {yova|soegaard}@di.ku.dk, alopez@inf.ed.ac.uk`

Abstract

Parsers are available for only a handful of the world's languages, since they require lots of training data. How far can we get with just a small amount of training data? We systematically compare a set of simple strategies for improving low-resource parsers: data augmentation, which has not been tested before; cross-lingual training; and transliteration. Experimenting on three typologically diverse low-resource languages—North Sámi, Galician, and Kazah—We find that (1) when only the low-resource treebank is available, data augmentation is very helpful; (2) when a related high-resource treebank is available, cross-lingual training is helpful and complements data augmentation; and (3) when the high-resource treebank uses a different writing system, transliteration into a shared orthographic spaces is also very helpful.

1 Introduction

Large annotated treebanks are available for only a tiny fraction of the world's languages, and there is a wealth of literature on strategies for parsing with few resources (Hwa et al., 2005; Zeman and Resnik, 2008; McDonald et al., 2011; Søgaard, 2011). A popular approach is to train a parser on a related high-resource language and adapt it to the low-resource language. This approach benefits from the availability of Universal Dependencies (UD; Nivre et al., 2016), prompting substantial research (Tiedemann and Agic, 2016; Agić, 2017; Rosa and Mareček, 2018), along with the VarDial and the CoNLL UD shared tasks (Zampieri et al., 2017; Zeman et al., 2017, 2018).

But low-resource parsing is still difficult. The organizers of the CoNLL 2018 UD shared task (Zeman et al., 2018) report that, in general, results on the task's nine low-resource treebanks "are extremely low and the outputs are hardly useful for

downstream applications." So if we want to build a parser in a language with few resources, what can we do? To answer this question, we systematically compare several practical strategies for low-resource parsing, asking:

1. What can we do with only a very small *target* treebank for a low-resource language?

2. What can we do if we also have a *source* treebank for a related high-resource language?

3. What if the source and target treebanks do not share a writing system?

Each of these scenarios requires different approaches. **Data augmentation** is applicable in all scenarios, and has proven useful for low-resource NLP in general (Fadaee et al., 2017; Bergmanis et al., 2017; Sahin and Steedman, 2018). Transfer learning via **cross-lingual training** is applicable in scenarios 2 and 3. Finally, **transliteration** may be useful in scenario 3.

To keep our scenarios as realistic as possible, we assume that no taggers are available since this would entail substantial annotation. Therefore, our neural parsing models must learn to parse from words or characters—that is, they must be **lexicalized**—even though there may be little shared vocabulary between source and target treebanks. While this may intuitively seem to make cross-lingual training difficult, recent results have shown that lexical parameter sharing on characters and words can in fact improve cross-lingual parsing (de Lhoneux et al., 2018); and that in some circumstances, a lexicalized parser can outperform a delexicalized one, even in a low-resource setting (Falenska and Çetinoğlu, 2017).

We experiment on three language pairs from different language families, in which the first of each is a genuinely low-resource language: North

Sámi and Finnish (Uralic); Galician and Portuguese (Romance); and Kazakh and Turkish (Turkic), which have different writing systems[1]. To avoid optimistic evaluation, we extensively experiment only with North Sámi, which we also analyse to understand *why* our cross-lingual training outperforms the other parsing strategies. We treat Galician and Kazakh as truly held-out, and test only our best methods on these languages. Our results show that:

1. When no source treebank is available, data augmentation is very helpful: dependency tree morphing improves labeled attachment score (LAS) by as much as 9.3%. Our analysis suggests that syntactic rather than lexical variation is most useful for data augmentation.

2. When a source treebank is available, cross-lingual parsing improves LAS up to 16.2%, but data augmentation still helps, by an additional 2.6%. Our analysis suggests that improvements from cross-lingual parsing occur because the parser learns syntactic regularities about word order, since it does not have access to POS and has little reusable information about word forms.

3. If source and target treebanks have different writing systems, transliterating them to a common orthography is very effective.

2   Methods

We describe three techniques for improving low-resource parsing: (1) two data augmentation methods which have not been applied before for dependency parsing, (2) cross-lingual training, and (3) transliteration.

2.1   Data augmentation by dependency tree morphing (Morph)

Sahin and Steedman (2018) introduce two operations to augment a dataset for low-resource POS tagging. Their method assumes access to a dependency tree, but they do not test it for dependency parsing, which we do here for the first time. The first operation, *cropping*, removes some parts of a sentence to create a smaller or simpler, meaningful

(a) Original sentence.

(b) Cropped sentence.

(c) Rotated sentence.

Figure 1: Examples of dependency tree morphing operations on the sentence "She wrote me a letter".

sentence. The second operation, *rotation*, keeps all the words in the sentence but re-orders subtrees attached to the *root* verb, in particular those attached by NSUBJ (nominal subject), OBJ (direct object), IOBJ (indirect object), or OBL (oblique nominal) dependencies. Figure 1 illustrates both operations.

It is important to note that while both operations change the set of words or the word order, they do not change the dependencies. The sentences themselves may be awkward or ill-formed, but the corresponding analyses are still likely to be correct, and thus beneficial for learning. This is because they provide the model with more examples of variations in argument structure (cropping) and in constituent order (rotation), which may benefit languages with flexible word order and rich morphology. Some of our low-resource languages have these properties—while North Sámi has a fixed word order (SVO), Galician and Kazakh have relatively free word order. All three languages use case marking on nouns, so word order may not be as important for correct attachment.

Both rotation and cropping can produce many trees. We use the default parameters given in (Sahin and Steedman, 2018).

---

[1] We select high-resource language based on language family, since it is the most straightforward way to define language relatedness. However, other measurement (e.g., WALS (Dryer and Haspelmath, 2013) properties) might be used.
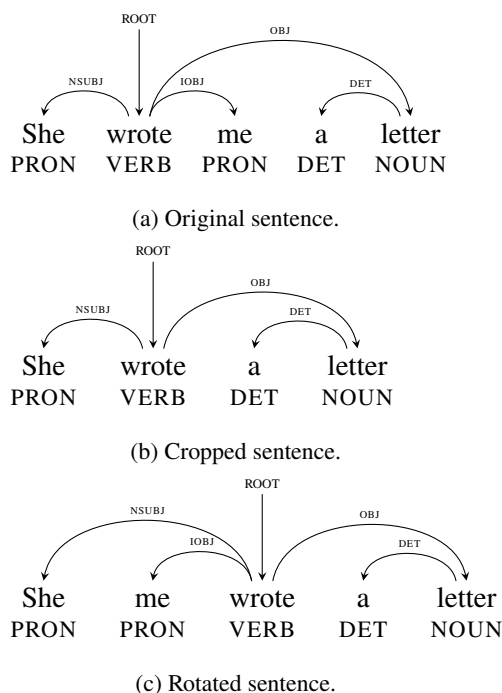
2.2 Data augmentation by nonce sentence generation (Nonce)

Our next data augmentation method is adapted from Gulordava et al. (2018). The main idea is to create *nonce* sentences by replacing some of the words which have the same syntactic annotations. For each training sentence, we replace each content word—nouns, verbs, or adjective—with an alternative word having the same universal POS, morphological features, and dependency label.[2] Specifically, for each content word, we first stochastically choose whether to replace it; then, if we have chosen to replace it, we uniformly sample the replacement word type meeting the corresponding constraints. For instance, given a sentence "*He borrowed a book from the library.*", we can generate the following sentences:

(1) a. He <u>bought</u> a book from the <u>shop</u> .

 b. He <u>wore</u> a <u>umbrella</u> from the library .

This generation method is only based on syntactic features (i.e., morphology and dependency labels), so it sometimes produces nonsensical sentences like 1b. But since we only replace words if they have the same morphological features and dependency label, this method preserves the original tree structures in the treebank. Following (Gulordava et al., 2018), we generate five nonce sentences for each original sentence.

2.3 Cross-lingual training

When a source treebank is available, model transfer is a viable option. We perform model transfer by cross-lingual parser training: we first train on both source and target treebanks to produce a single model, and then fine tune the model only on the target treebank. In our preliminary experiments (Appendix A), we found that fine tuning on the target treebank was effective in all settings, so we use it in all applicable experiments reported in this paper.

2.4 Transliteration

Two related languages might not share a writing system even when they belong to the same family. We evaluate whether a simple transliteration would be helpful for cross-lingual training in this case. In our study, the Turkish treebank is written in extended Latin while the Kazakh treebank is written in Cyrillic. This difference potentially makes model transfer less useful, and means we might not be able to leverage lexical similarities between the two languages. We pre-process both treebanks by transliterating them to the same "pivot" alphabet, basic Latin.[3]

The mapping from Turkish is straightforward. Its alphabet consists of 29 letters, 23 of which are in basic Latin. The other six letters, 'ç','ğ', 'ı', 'ö', 'ş', and 'ü', add diacritics to basic Latin characters, facilitating different pronunciations.[4] We map these to their basic Latin counterparts, e.g., 'ç' to 'c'. For Kazakh, we use a simple dictionary created by a Kazakh computational linguist to map each Cyrillic letter to the basic Latin alphabet.[5].

3 Experimental Setup

3.1 Dependency Parsing Model

We use the Uppsala parser, a transition-based neural dependency parser (de Lhoneux et al., 2017a,b; Kiperwasser and Goldberg, 2016). The parser uses an arc-hybrid transition system (Kuhlmann et al., 2011), extended with a static-dynamic oracle and SWAP transition to allow non-projective dependency trees (Nivre, 2009).

Let $w = w_0, \ldots, w_{|w|}$ be an input sentence of length $|w|$ and let w_0 represent an artificial ROOT token. We create a vector representation for each input token w_i by concatenating $(;)$ its word embedding, $\mathbf{e}_w(w_i)$ and its character-based word embedding, $\mathbf{e}_c(w_i)$:

$$\mathbf{x}_i = [\mathbf{e}_w(w_i); \mathbf{e}_c(w_i)] \quad (1)$$

Here, $\mathbf{e}_c(w_i)$ is the output of a *character-level* bidirectional LSTM (biLSTM) encoder run over the characters of w_i (Ling et al., 2015); this makes the model fully open-vocabulary, since it can produce representations for any character sequence. We then obtain a *context-sensitive* encoding $\mathbf{h}_i$ using a *word-level* biLSTM encoder:

$$\mathbf{h}_i = [\text{LSTM}_f(\mathbf{x}_{0:i}); \text{LSTM}_b(\mathbf{x}_{|w|:i})] \quad (2)$$

We then create a configuration by concatenating the encoding of a fixed number of words on the top of the stack and the beginning of the buffer. Given

---

[2]The dependency label constraint is new to this paper.

[3]Another possible pivot is phonemes (Tsvetkov et al., 2016). We leave this as future work.

[4]https://www.omniglot.com/writing/turkish.htm

[5]The mapping from Kazakh Cyrilic into basic Latin alphabet is provided in Appendix B

Language	Treebank ID	train	dev.	test
Finnish	fi_tdt	14981	1875	1555
North Sámi	sme_giella	1128	1129	865
Portuguese	pt_bosque	8329	560	477
Galician	gl_treegal	300	300	400
Turkish	tr_imst	3685	975	975
Kazakh	kk_ktb	15	16	1047

Table 1: Train/dev split used for each treebank.

	original	+Morph	+Nonce
$\mathcal{T}_{100}$	1128	7636	4934
$\mathcal{T}_{50}$	564	3838	2700
$\mathcal{T}_{10}$	141	854	661

Table 2: Number of North Sámi training sentences.

this configuration, we predict a transition and its arc label using a multi-layer perceptron (MLP). More details of the core parser can be found in de Lhoneux et al. (2017a,b).

3.2 Parameter sharing

To train cross-lingual models, we use the strategy of de Lhoneux et al. (2018) for parameter sharing, which uses *soft* sharing for word and character parameters, and *hard* sharing for the MLP parameters. Soft parameter sharing uses a language embedding, which, in theory, learns what parameters to share between the two languages. Let $\mathbf{c}_j$ be an embedding of character c_j in a token w_i from the treebank of language k, and let $\mathbf{l}_k$ be the language embedding. For sharing on characters, we concatenate character and language embedding: $[\mathbf{c}_j; \mathbf{l}_k]$ for input to the character-level biLSTM. Similarly, for input to the word-level biLSTM, we concatenate the language embedding to the word embedding, modifying Eq. 1 to

$$\mathbf{x}_i = [\mathbf{e}_w(w_i); \mathbf{e}_c(w_i); \mathbf{l}_k] \quad (3)$$

We use the default hyperparameters of de Lhoneux et al. (2018) in our experiments. We fine-tune each model by training it further only on the target treebank (Shi et al., 2016). We use early stopping based on Label Attachment Score (LAS) on development set.

3.3 Datasets

We use Universal Dependencies (UD) treebanks version 2.2 (Nivre et al., 2018). None of our target treebanks have a development set, so we generate new train/dev splits by 50:50 (Table 1). Having large development sets allow us to perform better analysis for this study.

4 Parsing North Sámi

North Sámi is our largest low-resource treebank, so we use it for a full evaluation and analysis of

different strategies before testing on the other languages. To understand the effect of target treebank size, we generate three datasets with different *training* sizes: $\mathcal{T}_{10}$ (~10%), $\mathcal{T}_{50}$ (~50%), and $\mathcal{T}_{100}$ (100%). Table 2 reports the number of training sentences after we augment the data using methods described in Section 2. We apply MORPH and NONCE separately to understand the effect of each method and to control the amount of noise in the augmented data.

We employ two baselines: a monolingual model (§3.1) and a cross-lingual model (§2.3), both *without* data augmentation. The monolingual model acts as a simple baseline, to resemble a situation when the target treebank does not have any source treebank (i.e., no available treebanks from related languages). The cross-lingual model serves as a strong baseline, simulating a case when there is a source treebank. We compare both baselines to models trained with MORPH and NONCE augmentation methods. Table 3 reports our results, and we review our motivating scenarios below.

Scenario 1: we only have a very small target treebank. In the monolingual experiments, we observe that both dependency tree morphing (MORPH) and nonce sentence generation (NONCE) improve performance, indicating the strong benefits of data augmentation when there are no other resources available except the target treebank itself. In particular, when the number of training data is the lowest ($\mathcal{T}_{10}$), data augmentations improves performance up to 9.3% LAS.

Scenario 2: a source treebank is available. We see that the cross-lingual training (cross-base) performs better than monolingual models even with augmentation. For the $\mathcal{T}_{10}$ setting, cross-base achieves almost twice as much as the monolingual baseline (mono-base). The benefits of data augmentation are less evident in the cross-lingual setting, but in the $\mathcal{T}_{10}$ scenario, data augmentation still clearly helps. Overall, cross-lingual combined with data augmentation yields the best result.

size	MONOLINGUAL			CROSS-LINGUAL		
	mono-base	+Morph	+Nonce	cross-base	+Morph	+Nonce
$\mathcal{T}_{100}$	53.3	56.0 (+3.3)	56.3 (+3.0)	61.3 (+8.0)	60.9 (+7.6)	**61.7 (+8.4)**
$\mathcal{T}_{50}$	42.5	46.6 (+4.1)	46.5 (+4.0)	52.0 (+9.5)	51.7 (+9.2)	**52.0 (+9.5)**
$\mathcal{T}_{10}$	18.5	27.1 (+8.6)	27.8 (+9.3)	34.7 (+16.2)	**37.3 (+18.8)**	35.4 (+16.9)

Table 3: LAS results on North Sámi development data. *mono-base* and *cross-base* are models without data augmentation. % improvements over *mono-base* shown in parentheses.

4.1 What is learned from Finnish?

Why do cross-lingual training and data augmentation help? To put this question in context, we first consider their relationship. Finnish and North Sámi are mutually unintelligible, but they are typologically similar: of the 49 (mostly syntactic) linguistic features annotated for North Sámi in the Word Atlas of Languages (WALS; Dryer and Haspelmath, 2013), Finnish shares the same values for 42 of them.[6] Despite this and their phylogenetic and geographical relatedness, they share very little vocabulary: only 6.5% of North Sámi tokens appear in Finnish data, and these words are either proper nouns or closed class words such as pronouns or conjunctions. However, both languages do share many character-trigrams (72.5%, token-level), especially on suffixes.

Now we turn to an analysis of the $\mathcal{T}_{10}$ data setting, where we see the largest gains for all methods.

4.2 Analysis of data augmentation

For dependency parsing, POS features are important because they can provide strong signals whether there exists dependency between two words in a given sentence. For example, *subject* and *object* dependencies often occur between a NOUN and a VERB, as can be seen in Fig. 1a. We investigate the extent to which data augmentation is useful for learning POS features, using diagnostic classifiers (Veldhoen et al., 2016; Adi et al., 2016; Shi et al., 2016) to probe our model representations. Our central question is: do the models learn useful representations of POS, despite having no direct access to it? And if so, is this helped by data augmentation?

After training each model, we freeze the parameters and generate *context-dependent* representa-

---
[6]There are 192 linguistic features in WALS, but only 49 are defined for North Sámi. These features are mostly syntactic, annotated within different areas such as morphology, phonology, nominal and verbal categories, and word order.

POS	%dev	baseline	%diff. with	
			+Morph	+Nonce
INTJ	0.1	0.0	20.0	20.0
PART	1.5	70.1	7.7	0.8
NUM	1.9	19.2	15.1	-4.1
ADP	1.9	15.7	24.5	19.7
SCONJ	2.4	57.8	5.9	7.6
AUX	3.2	26.3	27.2	-4.9
CCONJ	3.4	91.3	-0.8	-4.2
PROPN	4.7	5.9	5.9	-5.9
ADJ	6.5	12.7	3.8	0.2
ADV	9.0	42.9	11.8	11.5
PRON	13.4	63.2	5.4	-2.7
VERB	25.7	72.4	-6.2	-4.5
NOUN	26.4	67.0	8.6	13.2

Table 4: Results for the monolingual POS predictions, ordered by the frequency of each tag in the dev split (%dev). %diff shows the difference between each augmentation method and monolingual models.

tions (i.e., the output of *word-level* biLSTM, $\mathbf{h}_i$ in Eq. 2), for the training and development data. We then train a feed-forward neural network classifier to predict the POS tag of each word, using only the representation as input. To filter out the effect of cross-lingual training, we only analyze representations trained using the *monolingual* models. Our training and development data consists of 6321 and 7710 tokens, respectively. The percentage of OOV tokens is 40.5%.

Table 4 reports the POS prediction accuracy. We observe that representations generated with monolingual MORPH seem to learn better POS, for most of the tags. On the other hand, representations generated with monolingual NONCE sometimes produce lower accuracy on some tags; only on nouns the accuracy is better than monolingual MORPH. We hypothesize that this is because NONCE sometimes generates meaningless sentences which confuse the model. In parsing this effect is less apparent, mainly because monolingual NONCE has the poorest POS representation for infrequent tags (%dev), and better representa-

| | Top nearest Finnish words | |
North Sámi	char-level	word-level
borrat (VERB; eat)	*herrat* (NOUN; gentleman) *kerrat* (NOUN; time) *naurat* (VERB; laugh)	*käydä* (VERB; go) *otan* (VERB; take) *sain* (VERB; get)
veahki (NOUN; help)	*nuuhki* (VERB; sniff) *väki* (NOUN; power) *avarsi* (VERB; expand)	*tyhjäksi* (ADJ; empty) *johonki* (PRON; something) *lähtökohdaksi* (NOUN; basis)
divrras (ADJ; expensive)	*harras* (ADJ; devout) *reipas* (ADJ; brave) *sarjaporras* (NOUN; series)	*välttämätöntä* (ADJ; essential) *mahdollista* (ADJ; possible) *kilpailukykyisempi* (ADJ; competitive)

Table 5: Most similar Finnish words for each North Sámi word based on cosine similarity.

tion of nouns.

4.3 Effects of cross-lingual training

Next, we analyze the effect of cross-lingual training by comparing the monolingual baseline to the cross-lingual model with MORPH.

Cross-lingual representations. The fact that cross-lingual model improves parsing performance is interesting, since Finnish and North Sámi have so little common vocabulary. What linguistic knowledge is transferred through cross-lingual training? We analyze whether words with the same POS category from the source and target treebanks have similar representations. To do this, we analyze the *head predictions*, and collect North Sámi tokens for which only the cross-lingual model correctly predicts the headword.[7] For these words, we compare token-level representations of North Sámi *development* data to Finnish *training* data.

We ask the following questions: Given the representation of a North Sámi word, what is the Finnish word with the most similar representation? Do they share the same POS category? Information other than POS may very well be captured, but we expect that the representations will reflect similar POS since POS is highly revelant to parsing. We use *cosine distance* to measure similarity.

We look at four categories for which cross-lingual training substantial improves results on the development set: adjectives, nouns, pronouns, and verbs. We analyze representations generated by two layers of the model in §3.1: (1) the output of character-level biLSTM (char-level), $\mathbf{e}_c(w_i)$ and

POS	char-level (%)	word-level (%)
ADJ	12.1	37.1
NOUN	55.8	63.5
PRON	12.9	68.0
VERB	34.2	69.0

Table 6: # of North Sámi tokens for which the most similar Finnish word has the same POS.

(2) the output of word-level biLSTM (word-level), i.e., $\mathbf{h}_i$ in Eq. 2.

Table 5 shows examples of top three closest Finnish training words for a given North Sámi word. We observe that character-level representation focuses on orthographic similarity of suffixes, rather than POS. On the word-level representations, we find more cases when the top closest Finnish words have the same POS with the North Sámi word. In fact, when we compare the most similar Finnish word (Table 6) quantitatively, we find that the word-level representations of North Sámi are often similar to Finnish word with the same POS; the same trend does not hold for character-level representations. Since very few word tokens are shared, this suggests that improvements in cross-lingual training might simply be due to syntactic (i.e. word order) similarities between the two languages, captured in the dynamics of the biLSTM encoder—despite the fact that it knows very little about the North Sámi tokens themselves. The word-level representation has advantage over the char-level representation in the way that it has access to contextual information like word order, and it has knowledge about the other words in the sentence.

Head and label prediction. Lastly, we analyze the parsing performance of the monolingual com-

---

[7]Another possible way is to look at the label predictions. But since the monolingual baseline LAS is very low, we focus on the unlabeled attachment prediction since it is more accurate.
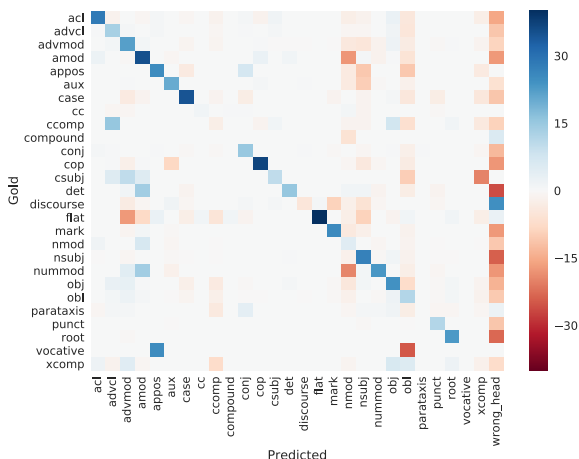
Figure 2: Differences between cross-lingual vs. monolingual confusion matrices. The last column represents cases of *incorrect* heads and the other columns represent cases for *correct* heads, i.e., each row summing to 100%. Blue cells show higher cross-lingual values and red cells show higher monolingual values.

pared to the cross-lingual models. Looking at the produced parse trees, one striking difference is that monolingual model sometimes predicts a "rootless" tree. That is, it fails to assign a head of any word with index '0' and label the dependency with a *root* label. In cases where the monolingual model predicts wrong parses and the cross-lingual model predicts the correct ones, we find that the "rootless" trees are predicted more than 50% of the time.[8] Meanwhile, the cross-lingual model learns to assign a word with head index '0', although sometimes it is the incorrect word (e.g., it is the second word, but the parser predicts the fifth word). This pattern suggests that more training examples at least helps the model to learn structural properties of a well-formed tree.

The ability of a parser to predict labels is contingent on its ability to predict heads, so we focus our analysis on two cases. How do monolingual and cross-lingual head prediction compare? And if both models predict the correct head, how do they compare on label prediction?

Figure 2 shows the *difference* between two confusion matrices: one for cross-lingual and one for monolingual models. The last column shows cases of *incorrect* heads and the other columns show label predictions when the heads are *correct*, i.e., each row summing to 100%. Here, blue cells

---

[8] The parsing model enforces the constraint that every tree should have a head, i.e., an arc pointing from a dummy root to a node in the tree. It does not, however, enforce that this arc be labeled *root*—the model must learn the labeling.

| | | CROSS-LINGUAL | |
Language	zero-shot	+fastText	+Morph
Galician	51.9	**72.8**	71.0
Kazakh	12.5	27.7	**28.4**
Kazakh (translit.)	21.2	31.1	**36.7**

Table 7: LAS results on **development sets**. *zero-shot* denotes results where we predict using model trained only on the source treebank.

highlight confusions that are more common for the cross-lingual model, while red cells highlight those more common for the monolingual model. For head prediction (last column), we observe that monolingual model makes higher errors especially for nominals and modifier words. In cases when both both models predict the correct heads, we observe that cross-lingual training gives further improvements in predicting most of the labels. In particular, regarding the "rootless" trees discussed before, we see evidence that cross-lingual training helps in predicting the correct root index, and the correct *root* label.

5 Parsing truly low-resource languages

Now we turn to two truly low-resource treebanks: Galician and Kazakh. These treebanks are most analogous to the North Sami $\mathcal{T}_{10}$ setting and therefore we apply the best approach, cross-lingual training with MORPH augmentation. Table 1 provides the statistics of the augmented data. For Galician, we use the Portuguese treebank as source while for Kazakh we use Turkish. Portuguese and Galician have high vocabulary overlap; 62.9% of Galician tokens appear in Portuguese data, while for Turkish and Kazakh they do not share vocabulary since they use different writing systems. However, after transliterating them into the same basic Latin alphabet, we observe that 9.5% of Kazakh tokens appear in the Turkish data. Both language pairs also share many (token-level) character trigrams: 96.0% for Galician-Portuguese and 66.3% for transliterated Kazakh-Turkish.

To compare our best approach, we create two baselines: (1) a pre-trained parsing model of the source treebank (zero-shot learning), and (2) a cross-lingual model initialized with *monolingual* pre-trained word embeddings. The first serves as a weak baseline, in a case where training on the target treebank is not possible (e.g., Kazakh only has 15 sentences for training). The latter serves as

	baseline	best system	CROSS-LINGUAL		
			+fastText	+Morph	rank
Galician	66.16	74.25	**70.46**	69.21	10/27
Kazakh (translit.)	24.21	31.93	25.28	**28.23**	2/27

Table 8: Comparison to CoNLL 2018 UD Shared Task on **test sets**. *best system* is the state-of-the-art model for each treebank: UDPipe-Future (Straka, 2018) for Galician and Uppsala (Smith et al., 2018) for Kazakh. *rank* shows our best model position in the shared task ranking for each treebank.

a strong baseline, in a case when we have access to pre-trained word embeddings, for the source and/or the target languages.

We treat a pre-trained word embedding as an external embedding, and concatenate it with the other representations, i.e., modifying Eq. 3 to $\mathbf{x}_i = [\mathbf{e}_w(w_i); \mathbf{e}_p(w_i); \mathbf{e}_c(w_i); \mathbf{l}_k]$, where $\mathbf{e}_p(w_i)$ represents a pre-trained word embedding of w_i, which we update during training. We use the pre-trained monolingual fastText embeddings (Bojanowski et al., 2017).[9] We concatenate the source and target pre-trained word embeddings.[10] For our experiments with transliteration (§2.4), we transliterate the entries of both the source and the target pre-trained word embeddings.

5.1 Experimental results

Table 7 reports the LAS performance on the development sets. MORPH augmentation improves performance over the zero-shot baseline and achieves comparable or better LAS with a cross-lingual model trained with pre-trained word embeddings.

Next, we look at the effects of transliteration (see Kazakh vs Kazakh (translit.) in Table 7). In the zero-shot experiments, simply mapping both Turkish and Kazakh characters to the Latin alphabet improves accuracy from 12.5 to 21.2 LAS. Cross-lingual training with MORPH further improves performance to 36.7 LAS.

5.2 Comparison with CoNLL 2018

To see how our best approach (i.e., cross-lingual model with MORPH augmentation) compares with the current state-of-the-art models, we compare it to the recent results from CoNLL 2018 shared task. Training state-of-the-art models may require lots of engineering and data resources. Our goal, however, is not to achieve the best performance,

but rather to systematically investigate how far simple approaches can take us. We report performance of the following: (1) the shared task baseline model (UDPipe v1.2; Straka and Straková, 2017) and (2) the best system for each treebank, (3) our best approach, and (4) a cross-lingual model with fastText embeddings.

Table 8 presents the overall comparison on the test sets. For each treebank, we apply the same sentence segmentation and tokenization used by each best system.[11] We see that our approach outperforms the baseline models on both languages. For Kazakh, our model (with transliteration) achieves a competitive LAS (28.23), which would be the second position in the shared task ranking. As comparison, the best system for Kazakh (Smith et al., 2018) trained a multitreebank model with four source treebanks, while we only use one source treebank. Their system use predicted POS as input, while ours depends solely on words and characters. The use of more treebanks and predicted POS is beyond the scope of our paper, but it is interesting that our approach can achieve the second best with such minimal resources. For Galician, our best approach outperforms baseline by 8.09 LAS points. Note that, Galician treebank does not come with training data. We use 50:50 train/dev split, while other teams might use higher split for training (for example, the best system (Straka, 2018) uses 90:10 train/dev split). Since we treat Galician as our test data, we did not tune on the proportion for training data, but we guess that this is the main reason why our system achieve rank 10 out of 27.

Compared to cross-lingual models with fastText embeddings (fastText vs. MORPH), we observe that our approach achieve better or comparable performance, showing its potential when there is

---

[9]The embeddings are available at https://fasttext.cc/docs/en/pretrained-vectors.html.

[10]If a word occurs in both source and target, we use the word embedding of the source language.

[11]UD shared task only provides unsegmented (i.e., sentence-level and token-level) raw test data. However, participants were allowed to use predicted segmentation and tokenization provided by the baseline UDPipe model.

not enough monolingual data available for training word embeddings.

6 Conclusions

In this paper, we investigated various low-resource parsing scenarios. We demonstrate that in the extremely low-resource setting, data augmentation improves parsing performance both in monolingual and cross-lingual settings. We also show that transfer learning is possible with *lexicalized* parsers. In addition, we show that transfer learning between two languages with different writing systems is possible, and future work should consider transliteration for other language pairs.

While we have not exhausted all the possible techniques (e.g., use of external resources (Rasooli and Collins, 2017; Rosa and Mareček, 2018), predicted POS (Ammar et al., 2016), multiple source treebanks (Lim et al., 2018; Stymne et al., 2018), among others), we show that simple methods which leverage the linguistic annotations in the treebank can improve low-resource parsing. Future work might explore different augmentation methods, such as the use of *synthetic* source treebanks (Wang and Eisner, 2018) or contextualized language model (Peters et al., 2018; Howard and Ruder, 2018; Devlin et al., 2018) for scoring the augmented data (e.g., using perplexity).

Finally, while the techniques presented in this paper might be applicable to other low-resource languages, we want to also highlight the importance of understanding the characteristics of languages being studied. For example, we showed that although North Sami and Finnish do not share vocabulary, cross-lingual training is still helpful because they share similar syntactic structures. Different language pairs might benefit from other types of similarity (e.g., morphological) and investigating this would be another interesting future work for low-resource dependency parsing.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *CoRR*, abs/1608.04207.

Željko Agić. 2017. Cross-Lingual Parser Selection for Low-Resource Languages. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 1–10. Association for Computational Linguistics.

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.

Toms Bergmanis, Katharina Kann, Hinrich Schütze, and Sharon Goldwater. 2017. Training data augmentation for low-resource morphological inflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 31–39. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

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

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573. Association for Computational Linguistics.

Agnieszka Falenska and Özlem Çetinoğlu. 2017. Lexicalized vs. delexicalized parsing in low-resource scenarios. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 18–24. Association for Computational Linguistics.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless Green Recurrent Networks Dream Hierarchically. In *Proceedings of the 2018 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1195–1205. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339. Association for Computational Linguistics.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Nat. Lang. Eng.*, 11(3):311–325.

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.

Marco Kuhlmann, Carlos Gómez-Rodrìguez, and Giorgio Satta. 2011. Dynamic Programming Algorithms for Transition-Based Dependency Parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 673–682. Association for Computational Linguistics.

Miryam de Lhoneux, Johannes Bjerva, Isabelle Augenstein, and Anders Søgaard. 2018. Parameter sharing between dependency parsers for related languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4992–4997. Association for Computational Linguistics.

Miryam de Lhoneux, Yan Shao, Ali Basirat, Eliyahu Kiperwasser, Sara Stymne, Yoav Goldberg, and Joakim Nivre. 2017a. From Raw Text to Universal Dependencies – Look, No Tags! In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies.*, Vancouver, Canada.

Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2017b. Arc-Hybrid Non-Projective Dependency Parsing with a Static-Dynamic Oracle. In *Proceedings of the The 15th International Conference on Parsing Technologies (IWPT).*, Pisa, Italy.

KyungTae Lim, Niko Partanen, and Thierry Poibeau. 2018. Multilingual dependency parsing for low-resource languages: Case studies on north saami and komi-zyrian. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. European Language Resource Association.

Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015. Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. In *Proceedings of the*

2015 Conference on Empirical Methods in Natural Language Processing, pages 1520–1530, Lisbon, Portugal. Association for Computational Linguistics.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-Source Transfer of Delexicalized Dependency Parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 62–72. Association for Computational Linguistics.

Joakim Nivre. 2009. Non-Projective Dependency Parsing in Expected Linear Time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 351–359. Association for Computational Linguistics.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, John Bauer, Sandra Bellato, Kepa Bengoetxea, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Radu Ion, Elena Irimia, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Tolga Kayadelen, Václava Kettnerová, Jesse Kirchner, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macke-

tanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Shinsuke Mori, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adédayọ̀ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Shoval Sadde, Shadi Saleh, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Yuta Takahashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Veronika Vincze, Lars Wallin, Jonathan North Washington, Seyi Williams, Mats Wirén, Tsegay Woldemariam, Tak-sum Wong, Chunxiao Yan, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. 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.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European

Language Resources Association (ELRA).

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Mohammad Sadegh Rasooli and Michael Collins. 2017. Cross-lingual syntactic transfer with limited resources. *Transactions of the Association for Computational Linguistics*, 5:279–293.

Rudolf Rosa and David Mareček. 2018. Cuni x-ling: Parsing under-resourced languages in conll 2018 ud shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 187–196. Association for Computational Linguistics.

Gozde Gul Sahin and Mark Steedman. 2018. Data Augmentation via Dependency Tree Morphing for Low-Resource Languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5004–5009. Association for Computational Linguistics.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural mt learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534. Association for Computational Linguistics.

Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018. 82 Treebanks, 34 Models: Universal Dependency Parsing with Multi-Treebank Models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123. Association for Computational Linguistics.

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*, pages 682–686. Association for Computational Linguistics.

Milan Straka. 2018. UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207. Association for Computational Linguistics.

Milan Straka and Jana Straková. 2017. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Sara Stymne, Miryam de Lhoneux, Aaron Smith, and Joakim Nivre. 2018. Parser Training with Heterogeneous Treebanks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 619–625. Association for Computational Linguistics.

Jörg Tiedemann and Željko Agic. 2016. Synthetic treebanking for cross-lingual dependency parsing. *J. Artif. Int. Res.*, 55(1):209–248.

Yulia Tsvetkov, Sunayana Sitaram, Manaal Faruqui, Guillaume Lample, Patrick Littell, David Mortensen, Alan W Black, Lori Levin, and Chris Dyer. 2016. Polyglot neural language models: A case study in cross-lingual phonetic representation learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1357–1366. Association for Computational Linguistics.

Sara Veldhoen, Dieuwke Hupkes, and Willem H. Zuidema. 2016. Diagnostic classifiers revealing how neural networks process hierarchical structure. In *CoCo@NIPS*.

Dingquan Wang and Jason Eisner. 2018. 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. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. 2017. Findings of the vardial evaluation campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 1–15. Association for Computational Linguistics.

Daniel Zeman, Jan Haji{č}, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21. Association for Computational Linguistics.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajic jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de-Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and

Josie Li. 2017. CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19. Association for Computational Linguistics.

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*.