Universal Dependencies are hard to parse – or are they?

Ines Rehbein[♣], **Julius Steen**[⋆], **Bich-Ngoc Do**[⋆], **Anette Frank**[⋆]
Leibniz ScienceCampus
Institut für Deutsche Sprache Mannheim[♣]
Universität Heidelberg[⋆]
Germany
{rehbein,steen,do,frank}@cl.uni-heidelberg.de

Abstract

Universal Dependency (UD) annotations, despite their usefulness for cross-lingual tasks and semantic applications, are not optimised for statistical parsing. In the paper, we ask what exactly causes the decrease in parsing accuracy when training a parser on UD-style annotations and whether the effect is similarly strong for all languages. We conduct a series of experiments where we systematically modify individual annotation decisions taken in the UD scheme and show that this results in an increased accuracy for most, but not for all languages. We show that the encoding in the UD scheme, in particular the decision to encode content words as heads, causes an increase in dependency length for nearly all treebanks and an increase in arc direction entropy for many languages, and evaluate the effect this has on parsing accuracy.

1 Introduction

Syntactic parsing, and in particular dependency parsing, is an important preprocessing step for many NLP applications. Many different parsing models are available for many different languages, and also a number of annotation schemes that differ with respect to the linguistic decisions they take. One of them is the Universal Dependencies (UD) scheme (Nivre et al., 2016) that has been developed to support cross-lingual parser transfer, and cross-lingual NLP tasks in general, and to provide a foundation for a sound cross-lingual evaluation.

While the value of the UD framework for multilingual applications is beyond doubt, it has been discussed that the annotation decisions taken in the UD framework are likely to decrease parsing accuracies, as most dependency-based parsers do prefer a chain representation of shorter dependencies over the UD-style encoding of dependencies where content words are heads, with function words attached as dependent nodes (*content-head* encoding). This is especially relevant for the encoding of coordinations, copula, and prepositions (Marneffe et al., 2014) (see figure 1). Several studies have addressed this problem and presented experiments on converted trees, offering evidence that a function-head encoding might increase the learnability of the annotation scheme (Schwartz et al., 2012; Popel et al., 2013; Silveira and Manning, 2015; Rosa, 2015; Versley and Kirilin, 2015; Kohita et al., 2017).

Evaluating the learnability of annotation frameworks, however, is not straightforward and attempts to do so have often resulted in an apples-to-oranges comparison as there are multiple factors that can impact parsing performance, including the language, the annotation scheme, the size of the treebank, and the parsing model. Even text-intrinsic properties such as domain and genre of the texts that are included in the treebank can influence results (Rehbein and van Genabith, 2007). It is not possible to control for all of them and this has made it extremely difficult to come to conclusions concerning the learnability of syntactic representations for different languages or annotation frameworks.

In the paper, we show that the design decisions taken in the UD framework have a negative impact on the learnability of the annotations for many languages, but not for all. We do this by evaluating three important design decisions made in the UD scheme and compare their impact on parsing accuracies for different languages.

The contributions of the paper are as follows. We test the claim that content-head dependencies are harder to parse, using three parsers that implement different parsing paradigms. We present a conversion algorithm that transforms the content-

Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017), pages 218-228,
Pisa, Italy, September 18-20 2017

head encoding of the UD treebanks for coordination, copula constructions and for prepositions into a function-head encoding and show that our conversion algorithm yields high accuracies (between 98.4% and 100%) for a back-and-forth conversion of *gold* trees.

We run parsing experiments on the original and the converted UD treebanks and compare the learnability of the annotations across 15 different languages, showing that language-specific properties play a cruicial role for the learning process. We further show that the changes in *dependency length* that result from the different encoding styles are *not* responsible for the changes in parsing accuracy.

The paper is structured as follows. We first review related work (§2) and present our conversion algorithm (§3). The data and setup for our experiments as well as the results are described in section §4. After a short discussion (§5) we conclude (§6).

2 Related work

It is well know from the literature that the linguistic framework used for a particular task has a great impact on the learnability of the annotations. Several studies have tried to evaluate and compare annotation schemes for syntactic parsing of one language (Kübler, 2005; Schwartz et al., 2012; Husain and Agrawal, 2012; Silveira and Manning, 2015) or across languages (Mareček et al., 2013; Rosa, 2015; Kohita et al., 2017), or have investigated the impact of a particular parsing model on the learnability of specific phenomena encoded in the framework (McDonald and Nivre, 2007; Goldberg and Elhadad, 2010).

Popel et al. (2013) present a thorough crosslingual investigation of different ways to encode coordination in a dependency framework. They did, however, not address the issue of learnability of the different encodings. This has been done in Maraček et al. (2013), who reach the somewhat disenchanted conclusion that the observed results of their experiments are "unconvincing and not very promising" (Mareček et al., 2013).

Versley and Kirilin (2015) look at the influence of languages and annotation schemes in universal dependency parsing, comparing 5 different parsers on 5 languages using two variants of UD schemes. They state that encoding content words as head has a negative impact on parsing results and that PP attachment errors account for a large portion of the differences in accuracy between the different parsers and between treebanks of varying sizes.

Recent work by Gulordava and Merlo (2016) has looked at word order variation and its impact on dependency parsing of 12 languages. They focus on word order freedom and dependency length as two properties of word order that systematically vary between different languages. To assess their impact on parsing accuracy, they modify the original treebanks by minimising the dependency lengths and the entropy of the head-direction (whether the head of dependent *dep* can be positioned to the left, to the right, or either way), thus creating *artificial* treebanks with systematically different word order properties. Parsing results on the modified treebanks confirm that a higher variation in word order and longer dependencies have a negative impact on parsing accuracies. These results, however, do not hold for all languages.[1]

The work of Gulordava and Merlo (2016) can not be used to compare the impact of different encoding schemes on the learnability of the annotations, as the modifications applied by the authors do result in *artificial* treebanks and cannot be traced back to specific design decisions, thus making the results hard to interpret for our purposes.

Kohita et al. (2017) overcome this problem by providing a conversion algorithm for the three functional labels *case, dep, mark* from the UD scheme. They convert the representations for those labels into function-head encodings and present parsing experiments on 19 treebanks from the UD project. Their results corroborate earlier findings and show that the conversions improve results for 16 out of 19 languages, using two graph-based parsers (MST and RBG) with default feature templates.

Our work is similar in spirit to the one of Kohita et al. (2017). We do, however, address partly different linguistic phenomena, namely the encoding of adpositions, copula verbs and coordinations. In contrast to Kohita et al. (2017), we do not back-transform the parser output but evaluate the converted trees against a converted version of the gold trees, as it has been shown that the back-conversion results in error propagation, which is reflected in lower parsing accuracies (Silveira and

[1] For German, for instance, word order variability seems to have a much stronger impact on parsing results while optimising dependency length resulted in a lower LAS.

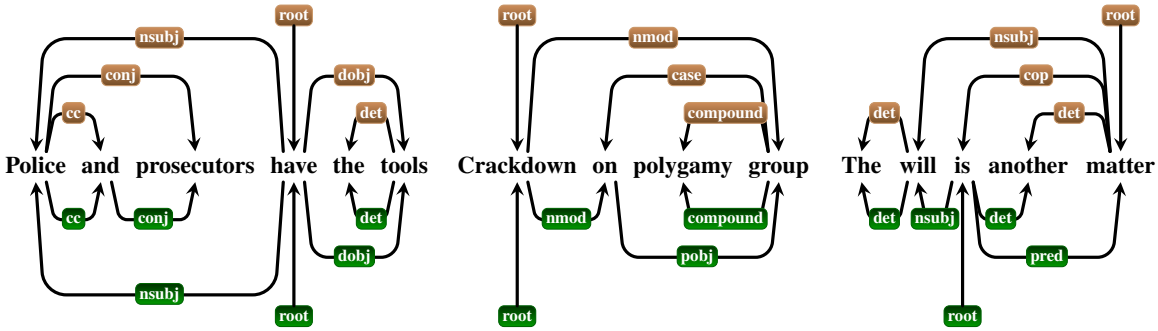

Figure 1: Dependency trees for conversion of coordination (left), prepositions (middle) and copula (right); UD encoding (brown, above) and modified trees with function words as heads (green, below).

Manning, 2015).[2]

Another difference to Kohita et al. (2017) concerns the parsers used in the experiments. While Kohita et al. (2017) use two graph-based parsing algorithms, we choose three parsers that represent different parsing paradigms, namely a transition-based parser, a graph-based parser and a head-selection parser. The latter is a neural parsing model that simply tries to find the best head for each token in the input. While the first two parsers use rich feature templates (and thus might be biased towards one particular encoding scheme), the head-selection parser does not use any pre-defined feature templates but learns all information directly from the input (§4.1).[3]

This allows us to test whether the previous results hold for parsers implementing different parsing paradigms and, crucially, whether they are independent of the feature templates used by the parsers. Finally, we are interested in the interaction between language, parser bias, and encoding scheme.

3 Conversion algorithm

The phenomena we consider in our experiments concern the encoding of copula verbs, coordinations and adpositions. All three address an important design decision taken in the UD project, namely to encode content words as heads.

We choose these because they are highly frequent in all the languages considered here and there is preliminary work discussing their impact on statistical parsing (Schwartz et al., 2012; Marneffe et al., 2014), claiming that encoding content words as heads has a negative impact on parsing accuracy, as has the UD way of encoding coordinations.

To compare the impact on parsing scores across different languages, we develop a conversion algorithm that transforms the original UD trees (figure 1, trees above) into a function-head style encoding (figure 1, trees below).[4] We first use our conversion algorithm to transform the encodings for individual constructions (**cop**ula, **prep**ositions, **coord**inations) and the combination of all the three (**c-p-c**) and then transform the converted trees back to the original encoding, using our conversion method. We then evaluate the trees that have been converted back and forth between UD style and function-head style against the original UD gold trees.

Table 1 shows results for a back-and-forth conversion of the original gold UD trees for 15 languages. Languages are ordered according to how many tokens in the test set are affected by the conversion. This ranges from 20.9% for Chinese (zh) to 45.7% for Farsi (fa), with an average of 34.7% over all 15 languages.[5] We can see that at least for gold trees, our conversion algorithm is able to transform between the two encodings without substantial loss of information.[6]

Errors in the back-conversion are partly due to inconsistencies in the annotations that are not always compliant with the UD scheme. Some of these issues have already been addressed in the

[2]The main goal of Kohita et al. (2017) was to increase parsing accuracy for UD parsing, thus making a back-conversion necessary. We, instead, are interested in a comparison of the learnability of the different schemes and thus can skip the back-conversion step.

[3]We do not use pretrained word embeddings in the experiments but learn the embeddings from the training data.

[4]Our code is available for download at http://wisscamp.de/en/research-2/resources.

[5]For comparison, the average ratio of converted tokens in the study of Kohita et al. (2017) is 6.3%.

[6]An exception is Farsi, where we observe a slightly higher LAS error rate, in particular for the conversion of coordinations.

			LAS				UAS	% affected
		size	**cop**	**prep**	**coord**	**c-p-c**	**c-p-c**	**c-p-c**
Chinese	**zh**	3,997	100.0	100.0	99.9	99.9	100.0	20.9
Estonian	**et**	14,510	99.9	100.0	100.0	99.9	100.0	23.6
Turkish	**tr**	3,948	99.9	99.8	99.8	99.4	99.8	27.9
Russian-SynTagRus	**ru**	48,171	100.0	100.0	100.0	100.0	100.0	30.6
German	**de**	14,118	99.8	100.0	99.8	99.6	100.0	33.2
Czech	**cs**	68,495	100.0	100.0	99.7	99.7	100.0	35.3
Romanian	**ro**	7,141	99.9	99.9	99.8	99.7	100.0	36.4
English	**en**	12,543	100.0	99.8	99.9	99.6	99.9	37.6
Croatian	**hr**	5,792	100.0	100.0	99.8	99.8	100.0	38.5
French	**fr**	14,554	100.0	99.8	99.9	99.8	99.9	38.5
Catalan	**ca**	13,123	99.9	99.5	99.9	99.4	99.8	38.8
Italian	**it**	12,837	100.0	100.0	99.9	100.0	100.0	40.3
Spanish	**es**	14,187	99.8	99.9	99.9	99.6	99.9	40.3
Bulgarian	**bg**	8,907	100.0	100.0	99.9	99.9	100.0	43.7
Farsi	**fa**	4,798	99.6	100.0	98.8	98.4	100.0	45.7
avg.		*16,475*	*99.9*	*99.9*	*99.8*	*99.6*	*99.9*	*35.4*

Table 1: LAS (excluding punctuation) on the test sets after round-trip conversion for individual transformations and for the combination of all (c-p-c: copula, prep, coord), evaluated against the original UD trees, and UAS for all conversions (c-p-c) (languages are ordered according to the amount of tokens affected by the combination of all conversions; zh: 20.9% – fa: 45.7%).

new release of the UD 2.0.[7] Other errors are due to language-specific constructions. A case in point are compositional preposition in Catalan (e.g. *per a*) where both parts are attached to the same head, while other sequences of prepositions have a chain-like attachment. Our conversion algorithm does not pay attention to language-specific properties that are neither encoded on the pos level nor in the dependency labels. It would, however, be straightforward to extend the algorithm to include these.

A final cause of errors in the back-conversion concerns coordinations with more than two conjuncts, where we have embedded coordinated constituents of the type *(A and B and (C and D))*. Here the back-conversion from the chain-like representation to UD looses information. In practice, however, these structures are not very frequent. For instance, in the English test set less than 0.8% of all sentences include a coordination of that particular type.

4 Experiments

We now want to use our conversion method to assess the impact of the content-head encoding in general and of individual, construction-specific encodings on parsing accuracies across different languages. In contrast to Kohita et al. (2017), our objective is *not* to improve UD parsing accuracies by using the conversion before parsing to increase the learnability of the representations and then convert the *parser output* back to the UD scheme. Our main goal is to use the conversion on *gold* trees in order to compare the impact it has for different languages and thus learn more about how to encode languages with different typological properties to improve monolingual dependency parsing results.

To rule out the influence of extrinsic factors such as data size or text type, we do not compare results across different treebanks and languages but modify specific annotation decisions and compare parsing accuracies for the original treebanks with the ones obtained on modified versions of the *same* treebank. Figure 1 illustrates the UD encoding (trees above) and the modified trees with function words as heads and a chain-like encoding of coordinations (trees below).

4.1 Data and setup

The data we use in our experiments comes from the UD treebanks (Nivre et al., 2016) v1.3. The selected 15 languages cover different language families and a range of typological properties. We

[7] The sixth release of the Universal Dependencies treebanks, v2.0, is available at http://universaldependencies.org.

		LAS			CNC		
		IMS	RBG	HSEL	IMS	RBG	HSEL
germanic	de	**84.3**	83.8	82.0	**79.7**	78.9	77.1
	en	**86.4**	86.3	86.0	**82.8**	82.2	82.3
iranian	fa	83.4	83.1	**83.9**	80.5	79.5	**80.8**
romance	ca	**89.5**	88.8	89.1	**84.0**	82.7	83.6
	es	**85.6**	85.2	85.2	**78.6**	77.5	78.0
	fr	**85.6**	84.4	85.2	**79.4**	77.6	78.6
	it	**89.6**	88.8	89.3	**84.3**	82.9	83.9
	ro	**79.9**	79.6	78.6	**75.4**	74.6	73.3
slavic	bg	**86.9**	84.9	85.6	**83.7**	80.8	81.7
	cs	**87.8**	86.1	85.7	**86.1**	83.9	83.5
	hr	79.9	**80.7**	78.1	77.2	**77.6**	74.9
	ru	**89.5**	**89.5**	86.8	**88.0**	87.8	84.4
sinitic	zh	**81.8**	79.4	80.4	**80.6**	77.9	79.1
finnic	et	**84.1**	83.9	75.3	**83.0**	82.6	73.0
turkic	tr	73.5	**75.1**	62.5	71.9	**73.4**	59.1

Table 2: LAS (excluding punctuation) and CNC (content dependencies only) on the test sets of the original treebanks.

choose three different non-projective parsers to assess the impact of specific parsing frameworks on the results, namely the graph-based RBG parser (Lei et al., 2014), the transition-based IMSTrans parser of Björkelund and Nivre (2015) (IMS), and our reimplementation of the head-selection parser of Zhang et al. (2017) (HSEL).

The RBG parser uses tensor decomposition and greedy decoding and the IMSTrans parser implements the (labeled) ArcStandard system, including a swap transition that can generate non-projective trees. The head-selection parser generates unlabeled trees by identifying the most probable head for each token in the input and then assigns labels to each head-dependent pair in a post-processing step. In contrast to the other two parsers, the head-selection parser does not use any predefined feature templates but selects the most probable head for each token based on word representations learned by a bidirectional long-short memory model (LSTM) (Hochreiter and Schmidhuber, 1997). Despite its simplicity and the lack of global optimisation, Zhang et al. (2017) report competetive results for English, Czech, and German.

For the first two parsers, we use default settings and the provided feature templates (for the RBG parser we use the *standard* setting *without* pretrained word embeddings), with no language-specific parameter optimisation.[8] We use the coarse-grained universal POS (Petrov et al., 2012) for all languages. The RBG and IMSTrans parser are trained on gold POS and morphological features provided by the UD project, the head-selection model is trained *without* morphological information, using word and POS embeddings only.

We choose the head-selection model to test whether a potential positive impact of the conversion might simply be a bias introduced by the feature templates, which might favour one particular encoding scheme. If we see the same improvements for all three parsers, we can be sure that the results are robust and not just an artefact of the feature templates used in the experiments.

For our experiments we systematically modify the input data and run parsing experiments on the original and on the converted treebanks. We have 15 settings per language (3 parsers x 5 treebank versions x 15 languages), which results in a total of 225 experiments. We hypothesize that the different modifications have a different effect on each language, which will be reflected in the changes in parsing accuracy when training and testing the parser on the different treebank versions.

4.2 Results for the original treebanks

Table 2 shows results for the three parsers on the original treebanks. We use the CNC metric proposed by Nivre (2016) and Nivre and Fang (2017) for UD evaluation. The metric excludes function words and punctuation from the evaluation and reports results only for *core* and *non-core* grammatical functions, thus providing a more informative and also more robust evaluation across different

[8] Please note that our goal is not to improve, or compare, results for individual languages but to assess the impact of different encoding decisions on the parsing accuracy for one language.

		IMS		RBG		HSEL	
	lang	CNC	Δ	CNC	Δ	CNC	Δ
ger	de	81.0	1.3	81.2	2.3	78.0	0.9
	en	83.6	0.8	83.4	1.2	83.6	1.3
ira	fa	84.2	3.7	83.4	3.9	83.6	2.8
rom	ca	85.6	1.6	85.0	2.3	84.9	1.3
	es	80.5	1.9	80.8	3.3	79.9	1.9
	fr	81.9	2.5	80.7	3.1	80.4	1.8
	it	86.1	1.8	86.1	3.2	85.5	1.6
	ro	75.7	0.3	75.3	0.7	73.6	0.3
sla	bg	85.4	1.7	83.8	3.0	83.8	2.1
	cs	87.3	1.2	85.2	1.3	84.2	0.7
	hr	77.4	0.2	77.3	-0.3	73.2	-1.7
	ru	89.2	1.2	88.7	0.9	82.1	-2.3
sin	zh	81.9	1.3	78.9	1.0	79.2	0.1
fin	et	84.4	1.4	82.8	0.2	74.7	1.7
tur	tr	71.6	-0.3	71.8	-1.6	58.3	-0.8

Table 3: CNC for the converted treebanks and differences Δ to the CNC obtained on the original treebanks.

languages.[9] Our evaluation does not provide a fair comparison between the parsers, as the different parsers do not have access to the same information (the head-selection parser, for instance, has no access to morphological information) and were not optimised for specific languages. Instead, our goal is to test whether the results of our conversion are robust across different languages and parsing models.

From the table we can see that the parsers perform differently well on the different treebanks. The transition-based parser provides best results for most languages and is only outperformed by the tensor-based RBG parser on Turkish (tr) and Croatian (hr) and by the head-selection parser on Farsi (fa), all three languages with rather small training sets.

It comes at not surprise that the head-selection parser, which has no access to morphological information or subword representations, has problems with Turkish (tr) and Estonian (et), which are both agglutinative languages. Despite the simplicity of the head-selection model, however, the parser produces competetive results for many languages and even outperforms the other two parsers on Farsi (fa).[10]

[9]Please note that the CNC metric considers the same number of tokens for evaluation in the original and converted treebanks, which is crucial for comparability.

[10]The head-selection model can easily be extended to include character-based embeddings or morphological embeddings, which will increase its performance on morphologically rich languages, but this is out of scope of the present study.

metric	orig	cop	prep	coord	c-p-c	Δ
Turkish						
with punc	77.4	76.9	76.6	76.7	76.4	-1.0
w/o punc	75.1	74.4	74.1	74.2	73.8	-1.3
CNC	73.4	72.9	72.6	71.9	71.8	-1.6
core	65.9	65.3	65.9	64.7	**67.1**	+1.2
non-core	75.5	74.9	74.4	73.9	73.2	-2.3
func	85.6	84.2	83.4	**88.2**	**86.0**	+0.4
Croatian						
with punc	80.2	78.7	79.4	**81.0**	80.1	-0.1
w/o punc	80.7	79.0	80.0	**81.5**	80.5	-0.2
CNC	77.7	75.5	76.9	**78.6**	77.3	-0.4
core	81.1	**81.5**	81.0	**81.7**	**81.9**	+0.7
non-core	76.8	74.0	75.9	**77.8**	76.1	-0.9
func	88.5	87.9	87.9	**89.1**	**88.7**	+0.2

Table 4: Results for different label sets for Turkish and Croatian (RBG parser) and difference (Δ) between original and converted treebank (**c**op-**p**rep-**c**oord).

4.3 Results for the converted treebanks

We now want to assess the impact of our conversions on the different languages. Table 3 shows CNC scores for the three parsers trained on the converted treebanks as well as the difference (Δ) to the results we get when training on the original treebanks.[11]

Our results confirm previous results from the literature (Schwartz et al., 2012; Marneffe et al., 2014) and show that our conversions are beneficial for nearly all languages. One exception is Turkish where CNC scores for all three parsers decrease. For Croatian, we observe only a minor increase for the IMSTrans parser and a decrease in results for the other two parsers.

To better understand the results for Turkish, we compare accuracies for the different label sets for the RBG parser which obtained best results on the Turkish treebank (Table 4). Most interestingly, we see that our conversions do indeed increase results for the core arguments (+1.2% labelled accuracy; improvements for csubj and ccomp) and also for the function tags (+0.4%), but all three conversions result in lower scores for the non-core dependency labels, especially for coordinations. These results highlight the importance of a detailed error analysis and show that overall parsing scores might be misleading.

Considering the small size of the Turkish treebank and the fact that the data has been converted automatically without manual correction, we can

[11]LAS and CNC scores for all parsers and each individual conversion are shown in table 7 in the appendix.

not rule out that the negative impact of the conversion on the non-core dependencies is merely an artefact of low data quality. This issue requires further investigation.

Looking at the results for Croatian, we see that the chain-like encoding of coordinations in our conversion experiments brings improvements for all subsets of grammatical functions. The other two conversions, however, result in a decrease in accuracy, which is also reflected in the results for the combined conversion (c-p-c). While for Turkish all three conversions on their own seem to decrease results and only the combination of all three converted encodings yields an improvement, for Croatian we get best results when changing the annotation of coordinations only and keeping the remaining representations in UD style. This increases CNC scores for RBG from 77.7% to 78.6% (+0.9). Our last finding suggests that a language-specific optimisation of annotation schemes for parsing might be worthwhile, and that there is a complex interaction between encoding styles, data properties (e.g. the size of the treebank) and language properties.

We also observe a correlation between language family and the degree to which the conversion improves performance. For all three parsers, we observe a similar ranking.[12] At the top is Farsi which benefits most from the conversion, while for Croatian and Turkish the results decrease. In general, the romance languages (fr, es, it, ca) seem to profit more from the transformations than the germanic and slavic languages. Romanian, however, an easter romance language, seems to behave different from the italo-western romance languages and shows only a slight increase in CNC.

In the next section, we turn to the question what it is that determines whether and how much a particular language will benefit from a specific choice of encoding. To that end, we focus on two language-specific properties, namely on dependency length and on the direction of the relations, i.e. head-initial versus head-final dependencies.

4.4 Dependency length

Previous work has discussed the different factors that might impact parsing accuracies across languages, such as word order properties, the high amount of unknown words for morphologically rich languages, ambiguity due to case syncretism, non-projectivity, ambiguity in head direction, and dependency length (Tsarfaty et al., 2010; Schwartz et al., 2012; Gulordava and Merlo, 2016).

	Lang	orig	cop	prep	coord	c-p-c
ger	de	3.4	0.98	1.01	1.03	1.03
	en	2.9	1.00	1.04	1.03	1.07
ira	fa	3.5	0.97	0.99	1.02	0.97
rom	ca	3.1	1.00	1.06	1.03	1.09
	es	2.8	0.99	1.07	1.04	1.11
	fr	2.8	0.99	1.07	1.03	1.09
	it	2.7	1.00	1.05	1.02	1.08
	ro	2.7	1.00	1.04	1.04	1.07
sla	bg	2.5	1.01	1.05	1.02	1.08
	cs	2.8	1.00	1.58	1.03	1.06
	hr	2.8	1.00	1.03	1.04	1.08
	ru	2.7	1.00	1.02	1.03	1.05
sin	zh	3.6	1.00	0.98	1.01	1.00
fin	et	2.6	1.00	1.00	1.03	1.02
tur	tr	2.6	1.00	1.01	1.01	1.02

Table 5: Avg. dependency length in the original treebank and DLM ratio for each modification

Gulordava and Merlo (2016) have investigated the influence of dependency length and arc direction entropy on parsing results, using *artificially created* treebanks. We adopt their measures to find out more about the impact of different encodings on natural languages. Following Gulordava and Merlo (2016), we compute the overall ratio of Dependency Length Minimisation (DLM) in the modified treebanks (as compared to the original treebanks), based on the data in the training set, as follows.

$$DLMRatio = \sum_{s} \frac{DL_s}{|s|^2} / \sum_{s} \frac{ModDL_s}{|s|^2} \quad (1)$$

The dependency length DL for each sentence s in the original treebank is calculated as the sum of the length of all arcs in the tree for sentence s,[13] and $ModDL$ refers to the dependency length in the modified treebank. A DLM ratio above 1 means that the treebank conversion resulted in a decreased dependency length in the data.[14]

[12] We obtain highly significant results for Spearman's rank correlation, computed on the differences Δ in CNC (see table 3), between all possible parser pairs (IMS-RBG, IMS-HSEL, RBG-HSEL) (all $p < 0.0006$).

[13] For the rightmost UD tree in Figure 1 DL_s is 7 while the length for the modified tree ($ModDL_s$) is 5.

[14] Please note that in contrast to Gulordava and Merlo (2016), who computed the DLM ratio between the original treebanks and an artificially created version of the same data where the order of the tokens had been modified, we compute the DLM ratio between two different encodings of the *same* data and thus their DLM ratios are not directly comparable to ours.

We can see that the modifications have quite a different effect on the average dependency length in the different treebanks (Table 5). While for many languages the combination of all modifications results in a minimisation of dependency length, this does not hold for Farsi and Chinese, and only slightly for Turkish, German and Estonian. It does not seem that the minimisation in dependency length is the responsible factor for the improvements in CNC. To test this, we fitted a linear regression model to the data and, as expected, did not find a significant correlation between dependency length and the changes in CNC accuracy for any of the parsers (IMSTrans: p=0.604, RBG: p=0.463, HSEL: p=0.943).[15]

We were thus not able to replicate the findings of Gulordava and Merlo (2016) who optimised UD trees for dependency length, thus generating artificial trees that were allowed to violate language-specific word order restrictions. They concluded that an increase in dependency length has, in general, a negative impact on parsing scores. This conclusion does not hold for our data. However, Gulordava and Merlo (2016) also found that minimising dependency length e.g. for German did not improve parsing accuracies the same way as it did for other languages.

Even if our conversion does result in a minimisation of dependency length in the treebanks, we conclude that the improvements in parsing accuracy are not due to the shorter dependencies. This raises the question what it is that makes the converted trees easier to learn and whether the differences are due to typological properties or merely reflect idiosyncrasies in the treebanks.

4.5 Arc direction entropy

We now look at the variation in the linear ordering between a head and its dependent as a potential factor that might impact parsing accuracy. Languages can be distinguished with regard to the proportion of head-initial versus head-final dependencies, which reflect typological differences between language families (Liu, 2010). Different treebank annotation schemes, however, can also influence the variation in arc direction, independent from the specific language of the treebank content.

To quantify this variation, we compute arc-direction entropy (ADE) (Gulordava and Merlo, 2016) in a treebank by iterating over all dependents in each individual arc and summing up the probability of the arc, represented by the POS of the dependent, the relation and the POS of the head, times the conditional entropy of the head direction, given the arc (Equation (2)).[16] An increase in ADE means that a particular modification introduced more variation with respect to the linear order of head and dependent for a specific relation.

$$H(Dir|Rel, H, D) = \sum_{rel,h,d} p(rel, h, d)H(Dir|rel, h, d) \quad (2)$$

For most languages, the conversion from content-head to function-head dependencies decreases ADE (Table 6). For some languages, we see a slight increase (Czech, Romanian, Catalan) while for Croatian, Russian and Turkish, the increase in entropy is substantial with 0.4 and 0.5, respectively. When fitting a linear regression model to the data, this time we see a significant effect on parsing accuracy (CNC) for the IMSTrans parser ($p = 0.01$) and the RBG parser ($p = 004$). For the head-selection parser, the correlation is even stronger with $p = 0.0002$.

We also experimented with lexicalised arc entropy but found no improvement over the unlexicalised model, probably due to data sparseness (see the discussion in Futrell et al. (2015)).

	lang	Δ cop	Δ prep	Δ coord	Δ c-p-c
ger	de	-0.26	-0.03	0.03	-0.23
	en	-0.56	-0.19	-0.01	-0.72
ira	fa	-0.73	0.07	0.02	-0.60
rom	ca	0.09	0.07	-0.01	0.16
	es	-0.19	-0.19	0.02	-0.36
	fr	-0.16	-0.15	0.04	-0.27
	it	-0.22	-0.11	0.02	-0.29
	ro	-0.13	0.17	0.04	0.09
sla	bg	-0.31	-0.10	0.05	-0.34
	cs	-0.30	0.20	0.07	0.03
	hr	0.16	0.21	0.03	0.41
	ru	0.17	0.19	0.05	0.41
sin	zh	-0.25	-0.00	0.03	-0.19
fin	et	-0.37	0.16	0.04	-0.16
tur	tr	0.19	0.28	0.03	0.50

Table 6: Difference (Δ) between avg. unlexicalised arc direction entropy (ADE) in the original treebank and in the modified treebanks

[15] We used R's lm function to predict the changes in CNC for each modified treebank version, based on the DLMratio.

[16] Futrell et al. (2015) discuss a methodological problem for using entropy for estimating word order properties, namely its sensitivity to sample size. We adress this by measuring variation in arc direction over n equally-sized random samples from each treebank (with replacement, $n = 1000$), and then report the average over all samples.

5 Discussion

Our findings suggest that it is not so much an increase in dependency length that goes along with the content-head representation implemented in the UD treebanks, but rather the increase in entropy for the position of the head that causes the loss in parsing accuracy when training a parser on UD-style dependencies.

Kohita et al. (2017) also discuss another property, namely the *head word vocabulary entropy*, as a potential factor that impacts parsing scores. Their measure is an implementation of an idea described in Schwartz et al. (2012). However, Kohita et al. (2017) did not observe a significant correlation between improvements in parsing accuracy (obtained by the RBG parser) and head word vocabulary entropy.

Our results show that the improvements we get through the conversion of content-head to function-head dependencies are not only due to the feature templates used by the parsers, which might introduce a bias towards one particular encoding, as we get similar improvements for the head-selection parser, a neural parser which does *not* use any predefined feature templates but learns its features directly from the input representations.

6 Conclusions

We presented a systematic investigation of the impact of specific annotation design decisions for statistical dependency parsing. We showed that claims that have been made for English (Schwartz et al., 2012) also hold for many other languages, but that the effect strength varies considerably.

We also showed that the UD encoding of adpositions, coordination and copula increases dependency length for all the languages we investigated except Persian and Chinese. This increase, however, does not directly translate to lower parsing scores. Head direction entropy, on the other hand, seems to have a stronger impact on parsing. This finding is consistent with the observations of Gulordava and Merlo (2016) obtained on *artificially* created data and their suggestion that at least for German, word order variability might have a higher impact on parsing difficulty than dependency length.

Finally, our results suggest that there is an interaction between typological properties and the effect strength of the improvments obtained by the treebank conversion. This provides interesting avenues for future research, as language generalisations might help us to design treebank encoding schemes that are optimised for specific languages, without having to repeat the same effort for each individual language.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and suggestions. This research has been conducted within the Leibniz Science Campus "Empirical Linguistics and Computational Modeling", funded by the Leibniz Association under grant no. SAS-2015-IDS-LWC and by the Ministry of Science, Research, and Art (MWK) of the state of Baden-Württemberg.

References

Anders Björkelund and Joakim Nivre. 2015. Non-deterministic oracles for unrestricted non-projective transition-based dependency parsing. In *Proceedings of the 14th International Conference on Parsing Technologies*, IWPT '15, pages 76–86, Bilbao, Spain, July.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Quantifying word order freedom in dependency corpora. In *Proceedings of the Third International Conference on Dependency Linguistics*, Depling 2015, pages 91–100.

Yoav Goldberg and Michael Elhadad. 2010. Inspecting the structural biases of dependency parsing algorithms. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 234–242, Uppsala, Sweden.

Kristina Gulordava and Paola Merlo. 2016. Multilingual dependency parsing evaluation: a large-scale analysis of word order properties using artificial data. *Transactions of the Association for Computational Linguistics*, 4:343–356.

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

Samar Husain and Bhasha Agrawal. 2012. Analyzing parser errors to improve parsing accuracy and to inform tree banking decisions. In *The 10th International Workshop on Treebanks and Linguistic Theories*, TLT.

Ryosuke Kohita, Hiroshi Noji, and Yuji Matsumoto. 2017. Multilingual back-and-forth conversion between content and function head for easy dependency parsing. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, EACL'17, pages 1–7, Valencia, Spain.

Sandra Kübler. 2005. How do treebank annotation schemes influence parsing results? Or how not to compare apples and oranges. In *Proceedings of Recent Advances in Natural Language Processing*, RANLP '05.

Tao Lei, Yu Xin, Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. 2014. Low-rank tensors for scoring dependency structures. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, ACL '14, pages 1381–1391.

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

David Mareček, Martin Popel, Loganathan Ramasamy, Jan Štěpánek, Daniel Zeman, Zdeněk Žabokrtský, and Jan Hajič. 2013. Cross-language study on influence of coordination style on dependency parsing performance. Technical report, UFAL.

Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies: a cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, LREC '14, Reykjavik, Iceland.

Ryan T. McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, EMNLP-CoNLL '07, pages 122–131, Prague, Czech Republic.

Joakim Nivre and Chiao-Ting Fang. 2017. Universal dependency evaluation. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies*, pages 86–95, Gothenburg, Sweden.

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 '16, Portorož, Slovenia.

Joakim Nivre. 2016. Universal dependency evaluation. Technical report, Uppsala University, Sweden.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, LREC '12, Istanbul, Turkey.

Martin Popel, David Marecek, Jan Stepánek, Daniel Zeman, and Zdenek Zabokrtský. 2013. Coordination structures in dependency treebanks. In *Annual Meeting of The European Chapter of The Association of Computational Linguistics*, EACL '13, pages 517–527.

Ines Rehbein and Josef van Genabith. 2007. Why is it so difficult to compare treebanks? TIGER and TüBa-D/Z revisited. In *The Sixth International Workshop on Treebanks and Linguistic Theories*, TLT '07, pages 115 – 126, Bergen, Norway.

Rudolf Rosa. 2015. Multi-source cross-lingual delexicalized parser transfer: Prague or Stanford? In *Proceedings of the Third International Conference on Dependency Linguistics*, DepLing '15, pages 281–290, Uppsala, Sweden.

Roy Schwartz, Omri Abend, and Ari Rappoport. 2012. Learnability-based syntactic annotation design. In *Proceedings of the 24th International Conference on Computational Linguistics*, COLING '12, pages 2405–2422, Mumbai, India.

Natalia Silveira and Christopher D. Manning. 2015. Does universal dependencies need a parsing representation? an investigation of english. In *Proceedings of the Third International Conference on Dependency Linguistics*, DepLing'15, pages 310–319, Uppsala, Sweden.

Reut Tsarfaty, Djame Seddah, Yoav Goldberg, Sandra Kübler, Marie Candito, Jennifer Foster, Yannick Versley, Ines Rehbein, and Lamia Tounsi. 2010. Statistical Parsing of Morphologically Rich Languages (SPMRL): What, How and Whither. In *Proceedings of the first workshop on Statistical Parsing of Morphologically Rich Languages*, SPMRL'10, Los Angeles, CA, USA.

Yannick Versley and Angelika Kirilin. 2015. What is hard in universal dependency parsing? In *The 6th Workshop on Statistical Parsing of Morphologically Rich Languages*, SPMRL '15.

Xingxing Zhang, Jianpeng Cheng, and Mirella Lapata. 2017. Dependency parsing as head selection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, EACL'17, pages 665–676, Valencia, Spain.

Appendix A. Supplemental Material

		IMS (LAS)					IMS (CNC)				
		orig	**cop**	**prep**	**coord**	**c-p-c**	**orig**	**cop**	**prep**	**coord**	**c-p-c**
germanic	de	84.3	85.0	84.3	84.9	**85.2**	79.7	80.9	79.9	80.5	**81.0**
	en	86.4	86.0	86.5	**87.0**	86.7	82.8	82.5	83.2	83.5	**83.6**
iranian	fa	83.4	85.6	84.6	84.5	**86.4**	80.5	83.0	81.9	82.0	**84.2**
romance	ca	89.5	89.5	89.2	**90.1**	89.9	84.0	84.3	84.1	85.3	**85.6**
	es	85.6	85.4	85.6	86.7	**86.8**	78.6	78.2	78.7	80.3	**80.5**
	fr	85.6	86.1	85.3	86.1	**87.0**	79.4	80.6	79.2	80.2	**81.9**
	it	89.6	89.9	90.1	**90.7**	90.5	84.3	85.0	85.1	86.0	**86.1**
	ro	79.9	79.4	79.6	**80.7**	80.0	75.4	74.8	75.1	**76.4**	75.7
slavic	bg	86.9	87.6	87.0	87.5	**88.0**	83.7	84.8	84.0	84.5	**85.4**
	cs	87.8	88.2	88.2	88.2	**88.8**	86.1	86.5	86.6	86.5	**87.3**
	hr	79.9	79.8	79.5	**82.2**	80.4	77.2	77.0	76.8	**79.3**	77.4
	ru	89.5	89.5	89.8	**90.6**	**90.6**	88.0	88.0	88.3	**89.2**	**89.2**
sinitic	zh	81.8	81.5	82.3	82.1	**82.9**	80.6	80.5	81.1	80.9	**81.9**
finnic	et	84.1	84.9	84.1	84.8	**85.5**	83.0	83.8	83.0	83.7	**84.4**
turkic	tr	73.5	73.8	**74.0**	73.3	73.6	71.9	72.3	**72.5**	71.1	71.6

		RBG (LAS)					RBG (CNC)				
		orig	**cop**	**prep**	**coord**	**c-p-c**	**orig**	**cop**	**prep**	**coord**	**c-p-c**
germanic	de	83.8	84.2	84.0	84.0	**85.4**	78.9	79.6	79.3	79.0	**81.2**
	en	86.3	86.0	86.2	86.4	**86.8**	82.2	82.1	82.5	82.5	**83.4**
iranian	fa	83.1	84.6	83.8	83.3	**86.1**	79.5	81.2	80.6	79.9	**83.4**
romance	ca	88.8	88.6	88.9	89.4	**89.6**	82.7	82.5	83.6	83.9	**85.0**
	es	85.2	85.4	85.9	85.8	**86.8**	77.5	77.9	78.9	79.0	**80.8**
	fr	84.4	85.1	84.8	85.6	**86.3**	77.6	78.9	78.5	78.8	**80.7**
	it	88.8	89.1	89.7	89.3	**90.8**	82.9	83.3	84.3	83.6	**86.1**
	ro	79.6	79.1	79.3	79.8	**79.9**	74.6	74.1	74.4	74.9	**75.3**
slavic	bg	84.9	85.2	85.5	85.3	**86.9**	80.8	81.4	81.8	81.4	**83.8**
	cs	86.1	86.0	86.3	85.9	**87.1**	83.9	83.9	84.2	83.8	**85.2**
	hr	80.7	79.0	80.0	**81.5**	80.5	77.7	75.5	76.9	**78.6**	77.3
	ru	89.5	88.8	89.4	90.0	**90.1**	87.8	87.1	87.8	88.3	**88.7**
sinitic	zh	79.4	78.7	79.6	78.6	**80.2**	77.9	77.3	78.4	77.0	**78.9**
finnic	et	83.9	83.3	83.4	**84.2**	84.1	82.6	81.9	82.2	**83.0**	82.8
turkic	tr	**75.1**	74.4	74.1	74.2	73.8	**73.4**	72.9	72.6	71.9	71.8

		HSEL (LAS)					HSEL (CNC)				
		orig	**cop**	**prep**	**coord**	**c-p-c**	**orig**	**cop**	**prep**	**coord**	**c-p-c**
germanic	de	82.0	82.6	82.2	82.5	**82.8**	77.1	**78.0**	77.2	77.6	**78.0**
	en	86.0	86.2	86.1	86.5	**86.8**	82.3	82.7	82.6	82.9	**83.6**
iranian	fa	83.9	85.2	84.3	84.3	**86.1**	80.8	82.4	81.3	81.2	**83.6**
romance	ca	89.1	89.4	89.1	**89.9**	89.6	83.6	84.1	83.8	**84.9**	**84.9**
	es	85.2	85.6	86.0	85.8	**86.3**	78.0	78.7	79.3	79.1	**79.9**
	fr	85.2	**86.2**	85.3	85.7	**86.2**	78.6	80.1	78.8	79.5	**80.4**
	it	89.3	89.5	89.4	89.7	**90.4**	83.9	84.0	83.8	84.3	**85.5**
	ro	78.6	78.2	78.1	**79.2**	78.7	73.3	73.2	72.7	**74.2**	73.6
slavic	bg	85.6	86.5	85.9	86.0	**87.0**	81.7	83.3	82.2	82.6	**83.8**
	cs	85.7	86.1	85.8	86.0	**86.5**	83.5	83.8	83.5	83.7	**84.2**
	hr	78.1	75.4	77.9	**79.6**	76.8	74.9	72.4	74.9	**76.7**	73.2
	ru	86.8	86.6	86.6	**87.6**	84.7	84.4	84.2	84.2	**85.2**	82.1
sinitic	zh	80.4	79.7	**80.7**	79.7	80.4	79.1	78.5	**79.4**	78.6	79.2
finnic	et	75.3	76.5	74.9	75.8	**77.0**	73.0	74.3	72.7	73.4	**74.7**
turkic	tr	**62.5**	61.7	62.3	62.3	62.2	**59.1**	58.4	59.0	58.2	58.3

Table 7: LAS (excluding punctuation) and CNC (content dependencies only) on the test sets for the original UD treebanks and for individual conversions (cop: copula, prep: prepositions, coord: coordination, c-p-c: combination of all three conversions).