Contribution of Typology to Universal Dependency Parsing

Anonymous ACL submission

Abstract

Universal Dependencies (UD) is a global initiative to create a standardized annotation for the dependency syntax of human languages. Addressing its deviation from typological principles, this study presents an empirical investigation of a typologically motivated transformation of UD proposed by William Croft. Our findings underscore the significance of the modifications across diverse languages, shedding light on the advantages and limitations of the transformations.

1 Introduction

Universal Dependencies (UD) (Nivre et al., 2016; de Marneffe et al., 2021) is widely used as a standard for morphosyntactic annotations. Ever since its initial release in October 2014, however, the scheme has been criticized with respect to its adherence to typological principles (Choi et al., 2021; Kanayama and Iwamoto, 2020). Croft et al. (2017) argue that the UD initiative, akin to prior parsing and tagging scheme proposals aimed at universal description of the world's languages, fails to refer explicitly to the extensive typological literature on universals, which accounts for the language-specific annotations that it provides besides those that are actually universal in typological terms. Therefore, they continue to propose their own dependency annotation scheme, claiming to represent cross-linguistic variations more comprehensively based on the following four design principles.

The first principle distinguishes constructions, which are universal, from strategies, which are language-specific, and favors classification based on the former. For example, a copula strategy, used in English to realize a predicate nominal construction, may be represented by a different strategy in another language, so the separate relation in UD for copulas, i.e., 'cop,' is absent in Croft et al. (2017)'s revision. The second principle emphasizes the use of the same labels for the same functions realized syntactically and morphologically, as in UD's replacement of the earlier dependency relations used to mark prepositional phrases, indicating a syntactic strategy, by the 'case' label, which is already used in these schemes to represent a morphological strategy with the same function. The third principle prioritizes information packaging over lexical semantics and contributes significantly to the provision of a more economic tag set, as in the substitution of the UD relations for different nominal modifiers with a single label, detailed in Section 3. The fourth principle emphasizes consideration of dependency structure ranks, including predicates, arguments, modifiers, and adverbs qualifying modifiers, as instantiated by Croft et al. (2017)'s different treatments of complex sentences, complex predicates, and arguments although they are all dependent on the predicate.

Croft et al. (2017) emphasize that the advantages brought about by their scheme may sacrifice the practical purposes pursued by UD, including achieving high parsing accuracy. However, they never admit that this is necessarily the case, and the exact state of affairs needs to be investigated through experimentation and empirical evaluation. This paper seeks to empirically investigate the impact of the proposed revisions on parsing accuracy. We hypothesize that it is more straightforward to parse treebanks with topologically informed UD annotation (referred to as TUD henceforth) than to parse ones with standard UD annotation. We expect significant but not necessarily fundamental improvement, as Croft et al. (2017)'s proposals address only the classification of dependency relations without affecting the overall tree structure.

2 Related Work

Some proposals address the typological limitations of UD through parsing architecture. Basirat and

Nivre (2021) integrate the notion of syntactic nuclei into the UD parsing framework to cope with the typological differences of languages. Their experimentation demonstrates that nucleus composition consistently improves parsing accuracy. This idea is further explored by Nivre et al. (2022), who find that the observed parsing improvement results from the greater capability of the enriched models of analyzing main predicates, nominal dependents, clausal dependents, and coordination structures.

Other proposals present alternative annotation schemes or revisions to UD. Gerdes et al. (2018) propose the Surface-Syntactic Universal Dependencies (SUD), claimed to be a richer and easier variant of UD. They argue that SUD treebanks enable cross-linguistic typological measures thanks to their distributional and functional criteria. Gerdes et al. (2019) recall the SUD's general principles, update its relation set, address annotation issues, and present an orthogonal layer of syntactic features. Gerdes et al. (2021) further suggest that a new treebank should initially be developed in SUD, even if a UD treebank is intended. The 2021 International Conference on Parsing Technologies (Oepen et al., 2021) Shared Task was dedicated to the additional structural layer of UD, known as Enhanced Universal Dependencies (EUD), used to encode grammatical relations that can be represented more adequately using graphical rather than purely rooted trees.

This paper examines an annotation scheme that can be regarded as a revised version of UD. Our modifications concern only the dependency labels and their scopes of application, while the head-dependent relations remain intact. This is because Croft et al. (2017) adopt the same dependency tree topology as that of UD, while they classify the dependencies differently in a number of cases. Furthermore, our conversion involves less radical dependency relation mappings and retains the majority of original UD labels regardless of the POS tags of the corresponding head and dependent tokens.

3 Transformation

One of the four ranks that Croft et al. (2017) enumerate for dependency structure is that of arguments, where they distinguish the subject relation from object and oblique. They label this relation 'sbj' regardless of its categorization as a noun phrase or a relative clause. This is realized in our script via the consolidation rules nsubj→sbj and csubj→sbj. Furthermore, they find it redundant to tag direct and indirect objects differently, so iobj→obj* and obj→obj* are included in our script as consolidation rules aimed to exclude 'iobj' from the list of dependency relations.

Croft et al. (2017) disagree on the dependency label distinction made in UD between complements in terms of grammatical role, including obligatory and nonobligatory control. Our consolidation rules ccomp→comp and xcomp→comp serve to neutralize the distinction. Moreover, they assert that UD treats resultatives as controlled complements, which it labels 'xcomp.' They suggest that these complex predicate elements should be labeled similarly to other secondary predicates and adverbs of manner, which are tagged 'sec.' The rule xcomp→sec is included to realize this. Thus, the fragmentation rules xcomp→comp and xcomp→sec have the same UD relation on their left-hand sides. xcomp→comp is set to apply where the POS tag of the token with the 'xcomp' dependency relation is VERB, which is assumed not to be the case for resultatives, where xcomp→sec is set to apply instead.

UD treebanks optionally set the type of an adverb as a morphological feature known as AdvType, with different values for adverbs of manner, location, time, quantity or degree, cause, and modal nature. On the other hand, Croft et al. (2017) propose that the diversity of adverbs in semantics, syntactic distribution, and morphological form needs to be captured and suggest that adverbs of manner should be labeled 'sec,' and ones expressing degree or hedging, aspect or modality, and location or time should be tagged 'qlfy,' 'aux,' and 'obl,' respectively. Therefore, the fragmentation rules advmod→sec, advmod→qlfy, advmod→aux*, and advmod→obl* are there to convert 'advmod' to each of the above relations if AdvType is set to the corresponding value. In cases where a different or no setting exists, advmod→obl* will apply by default, as Croft et al. (2017) assert that the UD 'advmod' relation should be excluded altogether.

Croft et al. (2017) analyze light verbs as complex predicates, tagged 'cxp,' unlike in UD, where they are treated similarly to nominal compounds. Therefore, the rule compound→cxp is included in our script to transform the UD compound relation to 'cxp' where the token's parent is POS tagged VERB, assumed to signal a light verb construction alongside the token's own compound dependency relation label. They also suggest that

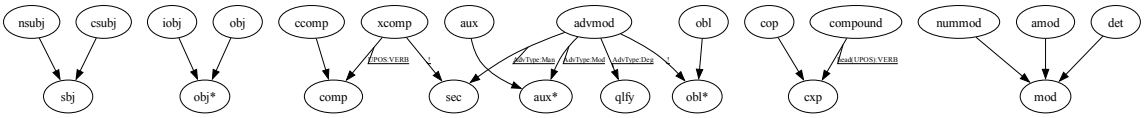

Figure 1: A summary of the transformation rules.

copulas should be treated as light verbs, hence the consolidation rule cop→cxp in our script. Furthermore, they suggest that 'nummod,' 'amod,' and 'det' should all be tagged 'mod,' as they involve the same type of information in general. The consolidation rules nummod→mod, amod→mod, and det→mod are there to realize this simplification. Figure 1 summarizes the transformations.

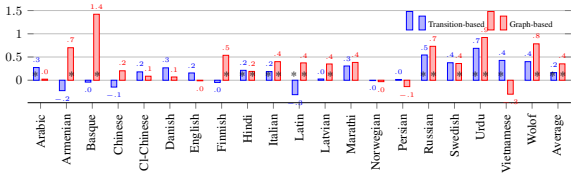

Figure 2: Absolute LAS improvement (or degradation). Significant results with p-value < 0.05 are marked.

4 Experiments and Results

We evaluate the impact of the typological transformations based on their contribution to the parsing performance. Our test benchmark consists of 20 treebanks from UD 2.12 belonging to diverse language families. In addition to the language diversity, we consider the presence of labels needed for the maximal application of the transformation rules. Table 1 outlines the selected treebanks with statistics about their sizes and transformed token ratios (Col. IR). Our analysis is based on the Labeled Attachment Score (LAS) obtained from two primary dependency parsing architectures: transition-based (Nivre, 2004) and graph-based parsing (McDonald et al., 2005). We use the UUParesr (de Lhoneux et al., 2017) for the former and the Biaffine parser (Dozat and Manning, 2017) for the latter with the settings outlined in Appendix A. We apply the transformation rules on each treebank and independently train three parsing models, each with distinct random seeds, using both the original (UD) and transformed treebanks (TUD). The average LASs on the development sets are reported in Cols. UD and TUD. Additionally, Col. Ora(cle) represents the upper bound for parsing performance, achievable if the dependency relations of the transformed tokens are predicted correctly. To assess the significance of differences between the UD and TUD results, we utilize McNemar's test, as detailed in Appendix B, and mark the significant differences (p-value $< .05$) with an asterisk.

First of all, the IR values indicate the importance of the typological transformation, applicable to almost 30% of the tokens, and that, if predicted correctly (Col. Ora), it can improve the performance by 2.1 and 3.0 points for the transition and graph-based parsing, respectively. However, the parsers can only harness a small but statistically significant portion of this potential improvement, with transition-based achieving 0.16 points and graph-based achieving 0.35 points. Figure 2 visualizes the absolute LAS improvement (or degradation) caused by the typological transformations. We can observe that, on most treebanks, the parsing models result in a better performance on typologically transformed treebanks and that, except for Latin, the negative results are statistically insignificant. These findings highlight the transformation's constructive role in enhancing parsing accuracy without introducing significant adverse effects.

Further investigation of the results reveals the varying contribution of the rules to the performance gain. Figure 3 illustrates the enhancement achieved by each transformation in classifying tokens that underwent the respective transformation. We can see that most rules constructively impact parsing with similar ranks for both parsers and that untransformed tokens (x→x) are not influenced. The most significant contribution arises from the consolidation rules. A crucial factor influencing their effectiveness is the inherent difficulty in distinguishing between source relations, often being misclassified as one another in UD, which is no longer an issue once they are merged in TUD. In particular, the effectiveness of the iobj→obj* rule is highlighted by the common misclassification scenario, where indirect objects ('iobj') are mistakenly identified as direct objects ('obj'). Therefore, the unification of 'iobj' and 'obj' prevents the parser from misclassi-

Language	Treebank	Family	Genus	Size	IR	Transition-based			Graph-based		
						UD	TUD	Ora	UD	TUD	Ora
Arabic	padt	Afro-Asiatic	Semitic	254K	0.20	77.83	78.10*	79.28	78.45	78.47	80.23
Armenian	armtdp	Indo-European	Indo-Iranian	47K	0.25	73.13	72.91	75.74	66.03	66.73*	70.88
Basque	bdt	Isolate		97K	0.26	74.94	74.90	76.87	67.74	69.16*	71.73
Chinese	gsd	Sino-Tibetan	Sinitic	111K	0.23	70.05	69.90	71.78	66.77	66.98	69.26
Classical Chinese	kyoto	Sino-Tibetan	Sinitic	406K	0.31	75.33	75.51	77.40	74.77	74.86	77.10
Danish	ddt	Indo-European	Germanic	91K	0.32	75.94	76.21	77.43	74.30	74.37	76.77
English	ewt	Indo-European	Germanic	230K	0.33	82.75	82.91	83.85	81.77	81.76	83.45
Finnish	tdt	Uralic	Finno-Ugric	181K	0.29	78.15	78.10	79.54	72.08	72.62*	74.72
Hindi	hdtb	Indo-European	Indo-Iranian	316K	0.22	87.58	87.79*	89.05	89.06	89.25*	90.67
Italian	isdt	Indo-European	Romance	288K	0.34	87.24	87.43*	88.26	87.15	87.55*	88.39
Latin	ittb	Indo-European	Italic	421K	0.33	83.26*	82.95	84.64	85.43	85.80*	87.10
Latvian	lvtb	Indo-European	Baltic	253K	0.29	79.81	79.83	81.48	78.14	78.49*	80.68
Marathi	ufal	Indo-European	Indo-Iranian	3K	0.30	48.71	49.01	57.31	49.77	50.15	59.82
Norwegian	bokmaal	Indo-European	Germanic	280K	0.31	87.42	87.41	88.34	88.12	88.09	89.21
Persian	seraji	Indo-European	Indo-Iranian	137K	0.26	81.26	81.27	82.63	78.47	78.33	80.40
Russian	taiga	Indo-European	Slavic	187K	0.28	64.95	65.50*	67.18	62.45	63.18*	65.50
Swedish	talbanken	Indo-European	Germanic	76K	0.34	76.02	76.40	78.21	70.71	71.07*	74.24
Urdu	udtb	Indo-European	Indo-Iranian	123K	0.24	76.19	76.87*	78.34	75.83	76.75*	78.66
Vietnamese	vtb	Austroasiatic	Vietic	46K	0.31	48.62	49.04*	52.75	47.55	47.25	51.79
Wolof	wtb	Niger-Congo	Atlantic-Congo	34K	0.28	72.02	72.42	73.93	67.04	67.83*	70.20
Summary				179K	0.28	75.06	75.22*	77.20	73.08	73.43*	76.04

Table 1: Average parsing accuracy (LAS) before (UD) and after (TUD) typological transformation.

fying them as each other. We found an analogous explanation for other consolidation rules that unify the clausal complements 'ccomp' and 'xcomp' into 'comp,' combine the subject relations 'nsubj' and 'csubj' into 'sbj,' and merge the determiner 'det' with modifiers 'amod' and 'nummod' into 'mod.' The small improvement made by cop→cxp in the transition-based parser is also due to the misclassification of copula as the compound, which is unified with copula in the typological scheme.

However, the fragmentation rules such as xcomp→sec and advmod→qlfy exhibit a negative influence. The detrimental impact of advmod→qlfy stems from the frequent mutual misclassification of adverbial and adjectival modifiers in UD, which persists even after typological transformation, manifested as mislabeling qualifying adverbs ('qlfy') as modifiers ('mod') in TUD, albeit at a higher rate, which is in turn because 'mod' in TUD has a broader scope than 'amod' in UD. In addition to the erroneous items present in both schemes, the rule introduces multiple frequent errors in TUD for tokens accurately classified in UD. The top four recurring errors include the misclassification of 'qlfy' as 'sbj' (13%), 'obl*' (12%), 'mod' (4%), and 'aux*' (4%) for tokens correctly classified in UD as 'advmod.' Similarly, the xcomp→sec rule negatively impacts parsing accuracy by misclassifying open clausal complements ('xcomp') and objects ('obj') in UD. This misclassification is

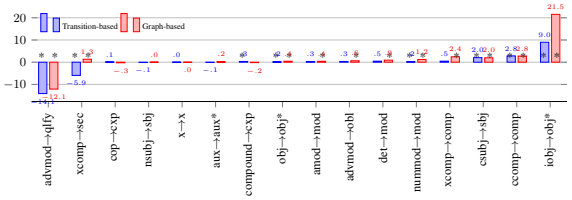

Figure 3: The transformation rules' contribution (or detraction). The results with p-value < 0.05 are marked.

due to their ambiguities and syntactic similarities, which persist between 'sec' and 'obj' in TUD, encompassing a large number of tokens, leading to increased errors. Putting it all together, we conclude that the fragmentation rules detract from parsing performance and that their degradation levels are proportional to the scales of their target relations.

Conclusion

The typological transformation of Universal Dependencies presents an advantage in terms of parsing performance. This benefit is observable across the two primary parsing approaches, namely the transition-based and the graph-based parsing, and in many languages. The positive impact on parsing performance can be attributed to the consolidation rules, which merge the dependency relation with similar typological properties. On the contrary, the parsing performance is hindered by fragmentation rules, indicating their detrimental effect in the context of Universal Dependencies.

5 Limitations

A limitation of this study is that not all of Croft et al. (2017)'s suggested transformation rules are considered due to a lack of annotation in the benchmark. Besides the labels on the right-hand sides of the rules in Section 3, Croft et al. (2017) name two tags for independent elements indicating indexation or agreement and linkers: 'idx' and 'lnk.' They categorize the above relations as common strategies, implying that they are not regarded as universal constructions. We have decided to ignore the above phenomena at this stage in the absence of clear clues as to how they are marked in each of the treebanks that contain them as independent tokens. We make the same decision for cases where it would be extremely difficult to identify the conditions for applying a rule, as in the case of depictives that are closely similar in structure to adverbial clauses. While these are both marked in UD as 'advcl,' Croft et al. (2017) suggest that the former should be labeled 'sec,' similarly to resultatives and manner adverbs, transformed via the consolidation rules xcomp→sec and advmod→sec, respectively. Our script, however, leaves 'advcl' tags unchanged, as one could hardly set proper conditions for an 'advcl'-to-'sec' transformation to apply, given the clues available on UD treebanks. In addition to these, our benchmark lacks any application for the rules advmod→sec and advmod→aux* due to the absence of optional morphological annotation in UD.

References

Ali Basirat and Joakim Nivre. 2021. Syntactic nuclei in dependency parsing – a multilingual exploration. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1376–1387, Online. Association for Computational Linguistics.

Hee-Soo Choi, Bruno Guillaume, and Karën Fort. 2021. Corpus-based language universals analysis using Universal Dependencies. In *Proceedings of the Second Workshop on Quantitative Syntax (Quasy, SyntaxFest 2021)*, pages 33–44, Sofia, Bulgaria. Association for Computational Linguistics.

William Croft, Dawn Nordquist, Michael Regan, and Katherine Looney. 2017. Linguistic typology meets Universal Dependencies. In *Proceedings of the 15th International Workshop on Treebanks and Linguistic Theories (TLT15)*, pages 63–75, Bloomington, IN. CEUR Workshop Proceedings.

Miryam de Lhoneux, Yan Shao, Ali Basirat, Eliyahu Kiperwasser, Sara Stymne, Yoav Goldberg, and Joakim Nivre. 2017. 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.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

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

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2018. SUD or surface-syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 66–74, Brussels, Belgium. Association for Computational Linguistics.

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2019. Improving surface-syntactic Universal Dependencies (SUD): MWEs and deep syntactic features. In *Proceedings of the 18th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2019)*, pages 126–132, Paris, France. Association for Computational Linguistics.

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2021. Starting a new treebank? go SUD! In *Proceedings of the Sixth International Conference on Dependency Linguistics (Depling, SyntaxFest 2021)*, pages 35–46, Sofia, Bulgaria. Association for Computational Linguistics.

Hiroshi Kanayama and Ran Iwamoto. 2020. How universal are Universal Dependencies? exploiting syntax for multilingual clause-level sentiment detection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4063–4073, Marseille, France. European Language Resources Association.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Joakim Nivre. 2004. Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 50–57, Barcelona, Spain. Association for Computational Linguistics.

Joakim Nivre, Ali Basirat, Luise Dürlich, and Adam Moss. 2022. Nucleus composition in transition-based dependency parsing. *Computational Linguistics*, 48(4):849–886.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, 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)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Stephan Oepen, Kenji Sagae, Reut Tsarfaty, Gosse Bouma, Djamé Seddah, and Daniel Zeman, editors. 2021. *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*. Association for Computational Linguistics, Online.

A Parsing Setup

Our transition-based parsing experiments utilize the implementation from Basirat and Nivre (2021), with the nucleus composition disabled.[1] For the graph-based experiments, we rely on the Biaffine module integrated into the SuPar parser.[2] In both parsers, we refrain from employing pre-trained embeddings, including both static and contextualized models, due to their inconsistent performance across different languages, which could potentially impact the research outcomes. Instead, we opt for a BiLSTM encoder in both scenarios to mitigate external influences and maintain result consistency. Neither do we employ any morphosyntactic features such as part-of-speech tags or morphological features to train the parsing models.

Both parsers are trained for 30 epochs with the word embedding size of 100 and the character embedding dimension of 100 for UUParser and 50 for SuPar. The UUParesr parameters are set to their default values as suggested by Nivre et al. (2022). The arc and relation MLP projection sizes of SuPar are set to 500 and 300, respectively, and the other parameters are set to their default values. We disable the projective parsing in both parsers.

The computational resource we use to train one transition-based model is a node of three CPUs and 5-10 GB memory in an HPC—however, the graph-based models, each consisting of 12M trainable parameters, are trained on NVIDIA Tesla V100 GPU.

[1] https://github.com/abasirat/uuparser
[2] https://github.com/yzhangcs/parser

Transformation		After (TUD)	
		1	0
Before (UD)	1	A	B
	0	C	D

Table 2: The contingency table for McNemar's test.

B Hypothesis Testing

We utilize McNemar's test to evaluate the significance of the parsing difference between the two schemes. McNemar's test is a paired-sample t-test for a dichotomous variable that takes two values. In our study, the dichotomous dependent variable of the test indicates whether a token is correctly classified in a scheme or not. The variable takes a value of 1 if the dependency head and label of a token are predicted accurately and a value of 0 otherwise. The categorical independent variable of the test refers to the two dependency schemes, UD and TUD. We collect the value of the dependent variable for all tokens across the two schemes, resulting in two lists of the size of the number of tokens, with the values in each list determining whether the token is classified correctly in the corresponding scheme or not. From these lists, we build a contingency table, shown in Table 2, with the following description:

- A: the number of tokens predicted correctly in both schemes
- B: the number of tokens predicted correctly in UD but incorrectly in TUD
- C: the number of tokens mispredicted in UD but predicted correctly in TUD
- D: the number of mispredicted tokens in both schemes.

With this setting, we estimate the p-value to reject the null hypothesis that the typological transformation does not impact parsing accuracy ($p_b = p_c$). We estimate the p-value based on the binomial distribution. To address the effect of randomness in the parsing models, we collect the statistics from the concatenation of the three runs with different random seeds.