

Replicating and Extending “*Because Their Treebanks Leak*”: Graph Isomorphism, Covariants, and Parser Performance

Mark Anderson

Universidade da Coruña, CITIC
Department of CS & IT
m.anderson@udc

Anders Søgaard

Dpt. of Computer Science
University of Copenhagen
soegaard@di.ku.dk

Carlos Gómez-Rodríguez

Universidade da Coruña, CITIC
Department of CS & IT
carlos.gomez@udc.es

Abstract

Søgaard (2020) obtained results suggesting the fraction of trees occurring in the test data isomorphic to trees in the training set accounts for a non-trivial variation in parser performance. Similar to other statistical analyses in NLP, the results were based on evaluating linear regressions. However, the study had methodological issues and was undertaken using a small sample size leading to unreliable results. We present a replication study in which we also bin sentences by length and find that only a small subset of sentences vary in performance with respect to graph isomorphism. Further, the correlation observed between parser performance and graph isomorphism in the wild disappears when controlling for covariants. However, in a controlled experiment, where covariants are kept fixed, we do observe a strong correlation. We suggest that conclusions drawn from statistical analyses like this need to be tempered and that controlled experiments can complement them by more readily teasing factors apart.

1 Introduction

We undertake a replication study of Søgaard (2020) which introduced graph isomorphism (DUG - directed unlabelled graph isomorphism) as a means of explaining differences in parser performance across different treebanks. It measures the ratio of graphs¹ in the test set that were also observed in the training data. It is intuitive that this would likely be related to parser performance.

However, DUG has two important covariants. The size of the training data impacts DUG because the smaller a treebank is, the less likely there will be many crossovers between training and test data. DUG is also tied to the mean sentence length in the test data: smaller sentences are much more likely to

¹Note that in the treebanks used in this paper, namely Universal Dependencies, well-formed trees are enforced.

have a tree structure already seen in the training, as there are fewer possible trees and the reverse is true for longer sentences, e.g. the number of possible trees for a sentence with 20 tokens is 12,826,228.

2 Related Work

There is a long history of investigating the causes of variance in parser performance. The effect of training data size on parser performance is well attested (Sagae et al., 2008; Falenska and Çetinoğlu, 2017; Strzyz et al., 2019; Dehouck et al., 2020). Sentence length has also been observed to impact performance (McDonald and Nivre, 2011). One likely factor behind this is different sentence lengths having different dependency distance distributions (Ferrer-i-Cancho and Liu, 2014) which in turn affects parsing as longer dependencies are typically harder to parse (Anderson and Gómez-Rodríguez, 2020; Falenska et al., 2020). Others have offered explanations based on linguistic characteristics such as morphological complexity (Dehouck and Denis, 2018; Çöltekin, 2020), part-of-speech bigram perplexity (Berdicevskis et al., 2018), and word order freedom (Gulordava and Merlo, 2016).

The history of reproduction and replication in NLP is not so well established, with only a few studies in recent years, e.g. on Universal Dependency (UD) parsing (Çöltekin, 2020) and on automatic essay scoring systems (Huber and Çöltekin, 2020).

Linear techniques, linear regression models or evaluating correlation coefficients are commonly used for statistical analyses of NLP systems. They have been used to model constituency parser performance (Ravi et al., 2008), to evaluate what affects annotation agreement (Bayerl and Paul, 2011), to investigate what impacts statistical MT systems (Guzman and Vogel, 2012), what impacts performance on span identifying tasks (Papay et al., 2020), and many other examples. Therefore, it is likely that lessons drawn from this replication

	Original			10 seeds		
	CoNLL18	UDPipe 1.2	UDPipe 2.0	CoNLL18	UDPipe 1.2	UDPipe 2.0
Training size	0.014	0.100	0.060	-0.019	-0.346	-0.005
+ DUG	0.228	0.061	0.097	-0.004	-0.553	0.091
+ $\langle L_{\text{test}} \rangle$	0.195	0.169	0.146	-0.007	-0.370	0.140
All	-0.078	0.157	0.086	-0.413	-0.138	0.106

Table 1: Issues with using multivariable linear model and cross-validation (CV) to evaluate explained variance. The first set of columns (Original) uses the exact same settings as the original paper (namely one CV split and the original seed) on the original data (CoNLL18) and the predictions from UDPipe 1.2 and UDPipe 2.0 for the extended data. The DUG explained variance is much smaller for the new data. The second set of columns show the same analysis but averaged over 10 different seeds used for the CV splits. The explained variances are almost all negative, which means the linear fit failed.

analysis will be impactful in a broader sense as the conclusions here can be applied in many sub-areas of NLP, namely the sensitive handling of covariants by using partial coefficients, controlled experiments, or signal subtraction; a strong adherence to visualising data; and considering whether the phenomena under consideration are likely to be sensitive to sentence length, as is often the case in NLP, and if so undertaking a sentence-length binning analysis to complement coarser analyses.

2.1 Original paper

Søgaard (2020) attempted to explain the difference of parser performance across treebanks by using DUG and also undirected unlabelled graph isomorphism (UUG). Two graphs are isomorphic if there is a renaming of vertices that makes them equal. The first process in calculating DUG (or UUG) is to collect the set of unique graphs that occur in the training data. In the original paper, this set of graphs is referred to as the isomorphisms. Once the training isomorphisms are obtained for a given treebank, the number of graphs in the test data that are members of one of these equivalence classes is counted. The final value is then the proportion of test instances that are isomorphic to the training data. This then gives a value between 0 (all test instances are unique) and 1 (no unique test instances).

The analysis was undertaken using a small sample of treebanks that were used at the CoNLL 2018 shared task, using the LAS of the top performing system for each treebank to measure parser performance (Zeman et al., 2018). The impact DUG (or UUG) has on parsing performance was evaluated by fitting a linear regression to the data with DUG as the control variable. A number of other potential measurements that could explain parser

performance were also taken into consideration, but only as alternative explanation and not covariants. The exception to this was using the size of the training data as a covariant. The explained variance and absolute error for each linear regression fit was reported using a three-fold cross-validation. The results suggested that DUG was the most strongly correlated measurement evaluated. We show that this result does not hold up when accounting for covariants, that using cross-validation method with the linear regression is not a robust method for an analysis like this, and that by controlling the main covariants of DUG, we can observe a more trustworthy correlation to parser performance.

3 Analysis and results

We evaluate directed graph isomorphism (DUG) as it was more strongly related to parser performance in the original paper.

Main covariants We focus on the two main covariants of DUG: training data size (in sentences) and mean sentence length of the test data, $\langle L_{\text{test}} \rangle$.

Data and parsers The data from the original paper consists of 33 UD treebanks, with LAS taken from the respective top performing parser from the CoNLL 2018 shared task (Zeman et al., 2018). Note that these systems are all variations of the biaffine graph-based parser of Dozat and Manning (2017). For replication, we also use a neural transition-based system UDPipe 1.2 (Straka et al., 2016), using UD models 2.4 and UD v2.5 (Zeman et al., 2019), and a neural graph-based system UDPipe 2.0 (Straka, 2018), using UD models 2.6 and UD v2.7 (Zeman et al., 2020). This results in 94 treebanks for UDPipe 1.2 and 90 for UDPipe 2.0. The difference is due to issues running the web-based UDPipe 2.0 on larger files.

3.1 Reproduction and replication

In the original paper, the analysis focuses on fitting a multi-variable linear regression to the data to control for covariants. However, the models only used training size plus one other variable as features. Further, cross-validation is used so as to avoid over-fitting. While over-fitting isn't directly an issue, the metrics that are typically reported overestimate the variance explained by a linear model, e.g. explained variance, η^2 , or R^2 (Lane et al., 2007). Averaging η^2 over different splits can potentially offset this positive bias but it requires a certain amount of data to be reliable. In Table 1, we show the results using the original data from Sogaard (2020). The values shown in the left-most column are exact reproductions of the original values. Only the value for $\langle L_{\text{test}} \rangle$ is different as the original paper appears to have used a normalised value. We also show η^2 for the linear model using all variables, which is negative, i.e. the fit failed.

We next show the results using UDPipe 1.2 and 2.0. While the values for training size on its own and with $\langle L_{\text{test}} \rangle$ are similar, the high η^2 for training size with DUG is no longer observed. This seems to be due to specious results born out of serendipitous splits for the smaller sample from CoNLL 2018.

We then tested this same procedure using different seeds to shuffle the cross-validation splits. The results are almost exclusively negative, i.e. the linear models failed to fit to the data at all. This further highlights an issue of using this methodology when sample size is small, as the random split can have large impact on the statistical metrics.

3.2 Extending the analysis

As the linear models performed so poorly, we measured the correlation coefficients (Spearman's ρ) for each of the variables with respect to LAS and also the potential covariants with respect to DUG. These are reported in Table 2 and we include visualisations of these in Figures 5 and 6 in the Appendix

	CoNLL18	UDPipe 1.2	UDPipe 2.0
size	0.46 (p=0.007)	0.54 (p<0.001)	0.37 (p<0.001)
DUG	-0.13 (p=0.458)	-0.13 (p=0.213)	-0.18 (p=0.083)
$\langle L_{\text{test}} \rangle$	0.20 (p=0.272)	0.35 (p=0.001)	0.33 (p=0.001)
size	0.44 (p=0.011)	0.42 (p<0.001)	0.46 (p<0.001)
$\langle L_{\text{test}} \rangle$	-0.96 (p<0.001)	-0.91 (p<0.001)	-0.92 (p<0.001)

Table 2: Spearman's ρ for variables with respect to LAS (top) and DUG (bottom).

	CoNLL18	UDPipe 1.2	UDPipe 2.0
log-size	0.055	0.319	0.126
+DUG	0.132	0.410	0.277
+$\langle L_{\text{test}} \rangle$	0.106	0.452	0.294
All	-0.184	0.412	0.229

Table 3: Using multivariable linear model and CV to evaluate explained variance with random shuffling (10 splits) and logarithmic transformation of treebank size.

for the CoNLL 2018 data and the UDPipe 2.0 data. Interestingly, DUG has the highest p-value for all systems, far from statistical significance. However, DUG appears to be strongly correlated to both covariants, especially $\langle L_{\text{test}} \rangle$ with $\rho > 0.9$ and $p < 0.001$ for all datasets and systems. Also of note is that training data size is convincingly correlated to LAS, but based on the linear models it doesn't appear to be predictive of parser performance. Based on this and on the visualisation of the data in Figures 5 and 6 in the Appendix (as well as visualisations of training size vs. LAS in the literature, see §2), it seems clear that the relation between these variables is not linear but logarithmic. We show LAS against training data size with a logarithmic scale in Figure 4 in the Appendix.

Table 3 shows the results of the limited linear model and cross-validation technique using 10 different seeds as above and using log training size. For these results, the explained variance of the models are all positive and relatively high, that is, the models manage to fit the data unlike in the original setup. This one change offsets the failure of the linear model technique, which is not surprising. However, it seems to suggest that DUG is not a useful feature, as training size with $\langle L_{\text{test}} \rangle$ outperforms training size with DUG for all datasets except CoNLL18. And the models which use all features are worse than just using training data size and $\langle L_{\text{test}} \rangle$, with the CoNLL18 model resulting in a negative explained variance, again meaning the fit failed. For CoNLL18, training data size and DUG does outperform the model using $\langle L_{\text{test}} \rangle$.

3.3 Sentence length binning

We analyse the relation between test sentence lengths and DUG by binning the data with respect to sentence length. This entails taking each sentence of length l for each treebank, in both the training and test data, and calculating DUG and the corresponding LAS based on these subsets. Figure 1 shows some of these bins (for sentences of

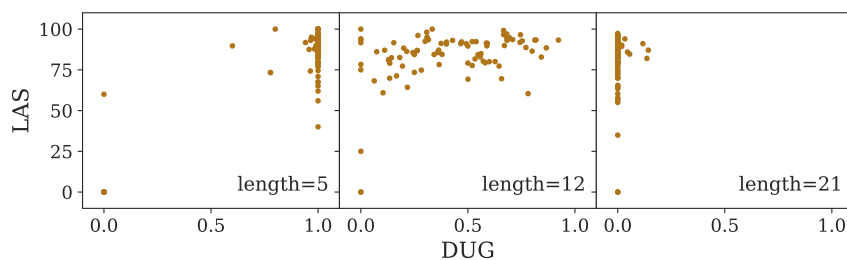


Figure 1: DUG binned wrt sentence length. Values are for UDPipe 2.0 with UD v2.7 for 90 treebanks.

length of 5, 12, and 21 tokens) for UDPipe 2.0. A full visualisation of each bin ranging from length 3 tokens to 30 is shown in Figure 7 in the Appendix.

DUG is almost exclusively 1.0 for shorter sentences, as can be seen in Figure 1 for sentence length 5. The number of possible directed trees for sentences with less tokens is too small for there not to be crossover: there are only 9 possible unlabelled trees for sentences of length 5 (Sloane, 1996). Conversely, for longer sentences, DUG is almost exclusively 0.0 as the number of possible tree structures is considerable (35,221,832 for sentences of length 21).

For a small subset of sentence lengths, ranging from length 9 to 14, there is meaningful spread of values for DUG, with a broadly-speaking linear relation with respect to LAS. Based on this result, i.e. that only certain sentence lengths are suitable for using DUG, we considered using a *focused* version of DUG, i.e. a variant calculated considering only sentences between length 9 and 14 in the training and test data. We then analysed how this measurement correlated with parser performance. Table 4 shows the correlations for focused DUG with respect to LAS, training size, and $\langle L_{\text{test}} \rangle$. While the correlation between focused DUG and LAS is much higher than for DUG and LAS, this is due to the focused version being much more strongly correlated to training size ($\rho = 0.91$ with a p-value

	UDPipe 1.2	UDPipe 2.0
LAS	0.47 (p<0.001)	0.31 (p=0.003)
size	0.91 (p<0.001)	0.91 (p<0.001)
$\langle L_{\text{test}} \rangle$	0.32 (p=0.002)	-0.34 (p=0.001)
log-size	0.319	0.126
+DUG	0.331	0.147
+$\langle L_{\text{test}} \rangle$	0.452	0.294
All	0.406	0.265

Table 4: Correlations wrt focused DUG (top) and explained variance (bottom) for focused DUG (sentence lengths 9 to 14) with shuffling for CV (10 seeds).

less than 0.001 for both datasets) and the correlation with $\langle L_{\text{test}} \rangle$ is much diminished. Also, this focused version of DUG improves performance for the linear model when used only with training data size, but $\langle L_{\text{test}} \rangle$ improves it much more. Using all 3 is again worse than just using training data size with $\langle L_{\text{test}} \rangle$, however, focused DUG doesn't lower the performance as much as the full variant does.

3.4 Controlling covariants

Having established that DUG does not improve linear models predicting LAS and that DUG is strongly correlated to training treebank size and $\langle L_{\text{test}} \rangle$, we attempted to find a signal by removing the background signals associated with these variables. We applied a linear fit to the training data size and LAS and then divided the LAS scores by the predicted values of that fit. Then we applied a linear fit to $\langle L_{\text{test}} \rangle$ and these *normalised* values and again divided these values out. Finally, we evaluated these *doubly normalised* values against DUG. This process is shown in Figure 2 for UDPipe 2.0 and the resulting coefficients for UDPipe 1.2 and 2.0 are in Table 7 of the Appendix. Removing the signals of the covariants results in a linear fit against DUG with a zero gradient and with a coefficient of 0.01 (p=0.926). Removing the variance associated with these covariants effectively removes any signal associated with DUG.

To corroborate this background subtraction analysis, we also report the partial coefficients in Table 5. When controlling for both covariants, correlations are small, and p-values very high, for both

	CoNLL18	UDPipe 1.2	UDPipe 2.0
DUG	-0.13 (p=0.458)	-0.13 (p=0.213)	-0.18 (p=0.083)
size	-0.44 (p=0.010)	-0.50 (p<0.001)	-0.46 (p<0.001)
$\langle L_{\text{test}} \rangle$	0.18 (p=0.329)	-0.13 (p=0.213)	0.21 (p=0.049)
both	-0.27 (p=0.126)	0.01 (p=0.915)	-0.12 (p=0.245)

Table 5: Partial Spearman's ρ for DUG with covariants.

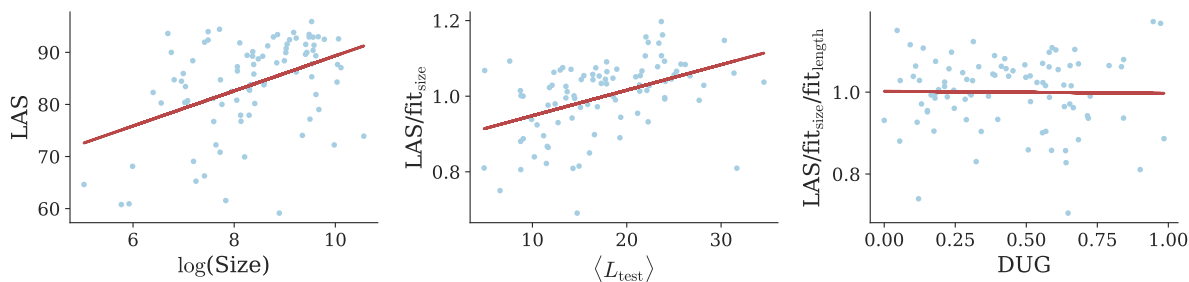


Figure 2: Visualisation of removing background signal associated with covariants of the log of training size ($\log(\text{Size})$) and mean test length $\langle L_{\text{test}} \rangle$. The spearman’s ρ for DUG and LAS is -0.18 ($p=0.083$), for DUG and $\text{LAS}/\text{bcg}_{\text{size}}$ is -0.40 ($p<0.001$) compared to $\langle L_{\text{test}} \rangle$ and $\text{LAS}/\text{bcg}_{\text{size}}$ of 0.465 ($p<0.001$), and finally DUG and $\text{LAS}/\text{bcg}_{\text{size}}/\text{bcg}_{L_{\text{test}}}$ is 0.01 ($p=0.926$).

UDPipe systems. CoNLL18 has a stronger signal, but it is negative (which is the opposite relation one would expect) and has a large p-value.

3.5 Controlled experiment - fixing covariants

We also evaluated DUG’s relation to LAS in a controlled experiment where we sampled subsets of treebanks keeping training data size constant and also the sentence length of both training and test data. We trained UDPipe 1.2 models (UDPipe 2.0 is not available beyond using pre-existing models), using standard settings. We were limited to 9 treebanks, as we required a reasonable amount of data and using only one sentence length reduces the number of usable treebanks. We combined all of the data for treebanks which had over 1200 sentences of length 12. We then created splits such that a single 1000-sentence training set was created by randomly sampling sentences. Then a number of 200-sentence test sets were created, generating as many splits as the data allowed for a given treebank.

$N_{\text{train_trees}}=1000, N_{\text{test_trees}}=200, \text{Sentence Length}=12$

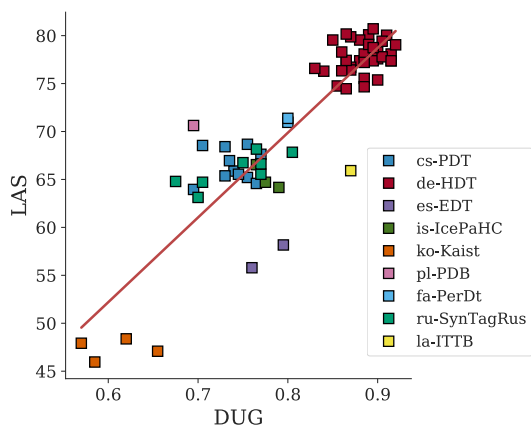


Figure 3: DUG vs LAS for controlled experiment. $\rho = 0.82$ ($p< 0.001$).

In this way we varied DUG indirectly, but by using different treebanks to sample from we obtained values spanning a reasonable range (0.6 - 0.9). This results in a Spearman’s ρ of 0.82 ($p<0.001$) and is visualised in Figure 3 in the Appendix. So in this rigid context, we do observe a very strong correlation between DUG and LAS, echoing the analysis from the sentence-length binning procedure.

4 Conclusion

With this case study we have shown the value of replicating analyses in NLP. Our analysis has shown that the original results were unreliable and it has highlighted methodological issues the original analysis had. Also, the results regarding the methodology presented here (i.e. the need to visualise and evaluate correlations before considering linear regression techniques, the potential sensitivity to sentence length of measurements used in NLP statistical analyses, the need to control for all covariants and evaluate their impact using partial coefficients at the very least, and finally that using controlled experiments can help better evaluate the impact of specific measurements and can complement statistical analyses) will likely be useful for other statistical analyses in different areas of NLP.

Acknowledgements

MA and CGR received funding from the European Research Council, under the EU’s Horizon 2020 research and innovation programme (FASTPARSE, grant agreement No 714150), from MINECO (ANSWER-ASAP, TIN2017-85160-C2-1-R), from Xunta de Galicia (ED431C 2020/11), and from CITIC, funded by Xunta de Galicia and the European Union (ERDF - Galicia 2014-2020 Program), by grant ED431G 2019/01. AS received funding from a Google Focused Research Award.

References

- Mark Anderson and Carlos Gómez-Rodríguez. 2020. **Inherent dependency displacement bias of transition-based algorithms**. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5147–5155, Marseille, France. European Language Resources Association.
- Petra Saskia Bayerl and Karsten Ingmar Paul. 2011. **What determines inter-coder agreement in manual annotations? A meta-analytic investigation**. *Computational Linguistics*, 37(4):699–725.
- Aleksandrs Berdicevskis, Çağrı Çöltekin, Katharina Ehret, Kilu von Prince, Daniel Ross, Bill Thompson, Chunxiao Yan, Vera Demberg, Gary Lupyan, Taraka Rama, and Christian Bentz. 2018. Using Universal Dependencies in cross-linguistic complexity research. In *Second Workshop on Universal Dependencies*, pages 8–17, Brussels, Belgium.
- Çağrı Çöltekin. 2020. **Verification, reproduction and replication of NLP experiments: A case study on parsing Universal Dependencies**. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 46–56, Barcelona, Spain (Online). Association for Computational Linguistics.
- Mathieu Dehouck, Mark Anderson, and Carlos Gómez-Rodríguez. 2020. **Efficient EUD parsing**. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 192–205, Online. Association for Computational Linguistics.
- Mathieu Dehouck and Pascal Denis. 2018. **A framework for understanding the role of morphology in Universal Dependency parsing**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2864–2870, Brussels, Belgium. Association for Computational Linguistics.
- Timothy Dozat and Christopher D Manning. 2017. **Deep biaffine attention for neural dependency parsing**. *Proceedings of the 5th International Conference on Learning Representations*.
- Agnieszka Falenska, Anders Björkelund, and Jonas Kuhn. 2020. **Integrating graph-based and transition-based dependency parsers in the deep contextualized era**. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 25–39, Online. 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, Pisa, Italy. Association for Computational Linguistics.
- Ramon Ferrer-i-Cancho and Haitao Liu. 2014. **The risks of mixing dependency lengths from sequences of different length**. *Glottometry*, 5(2):143–155.
- 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.
- Francisco Guzman and Stephan Vogel. 2012. **Understanding the performance of statistical MT systems: A linear regression framework**. In *Proceedings of COLING 2012*, pages 1029–1044, Mumbai, India. The COLING 2012 Organizing Committee.
- Eva Huber and Çağrı Çöltekin. 2020. **Reproduction and replication: A case study with automatic essay scoring**. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5603–5613, Marseille, France. European Language Resources Association.
- David M. Lane, David Scott, Mikki Hebl, Rudy Guerl, Dan Osherson, Heidi Zimmer, et al. 2007. *Online Statistics Education: A Multimedia Course of Study*. Online, Rice University, University of Houston Clear Lake, and Tufts University.
- Ryan McDonald and Joakim Nivre. 2011. **Analyzing and integrating dependency parsers**. *Computational Linguistics*, 37(1):197–230.
- Sean Papay, Roman Klinger, and Sebastian Padó. 2020. **Dissecting span identification tasks with performance prediction**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4881–4895, Online. Association for Computational Linguistics.
- Sujith Ravi, Kevin Knight, and Radu Soricut. 2008. **Automatic prediction of parser accuracy**. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 887–896, Honolulu, Hawaii. Association for Computational Linguistics.
- Kenji Sagae, Yusuke Miyao, Rune Saetre, and Jun’ichi Tsujii. 2008. **Evaluating the effects of treebank size in a practical application for parsing**. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 14–20, Columbus, Ohio. Association for Computational Linguistics.
- Neil James Alexander Sloane. 1996. The On-Line Encyclopedia of Integer Sequences. A000081.
- Anders Søgaard. 2020. **Some languages seem easier to parse because their treebanks leak**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2765–2770, Online. Association for Computational Linguistics.

Milan Straka. 2018. Udpipes 2.0 prototype at CoNLL 2018 UD shared task. *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207.

Milan Straka, Jan Hajič, and Jana Straková. 2016. Udpipes: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 4290–4297.

Michalina Strzyż, David Vilares, and Carlos Gómez-Rodríguez. 2019. [Viable dependency parsing as sequence labeling](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 717–723, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniel Zeman, Jan Hajič, 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, Brussels, Belgium. Association for Computational Linguistics.

Daniel Zeman, Joakim Nivre, et al. 2019. [Universal Dependencies 2.5](#). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Daniel Zeman, Joakim Nivre, et al. 2020. [Universal Dependencies 2.7](#). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

A Appendix

The appendix mainly consists of visualisations corresponding to the statistical analyses described in

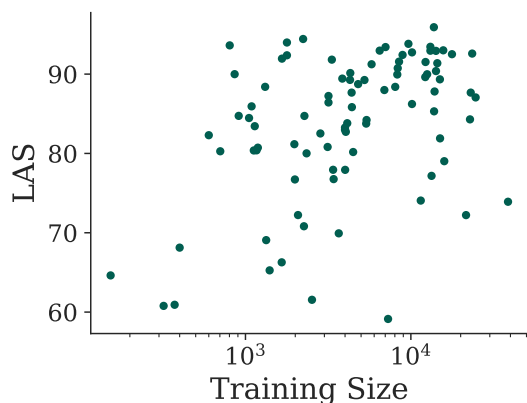


Figure 4: LAS with respect to training set size, in logarithmic scale, for UDPipe 2.0 and UD v2.7.

	UDPipe 1.2	UDPipe 2.0
DUG	0.47 (p<0.001)	0.31 (p=0.003)
size	-0.15 (p=0.153)	-0.10 (p=0.335)
$\langle L_{\text{test}} \rangle$	0.64 (p<0.001)	0.48 (p<0.001)
both	0.17 (p=0.110)	0.04 (p=0.683)

Table 6: Partial Spearman’s ρ for focused DUG (i.e. using only the measurement for sentences of length 9 to 14) with covariants.

the main body. Some additional information is given to supplement the main analyses in Tables 6 and 7 which give the correlations for the focused DUG analysis and the background removal process, respectively.

Figure 4 shows the logarithmic relation between LAS and the training data size for UDPipe 2.0 and UD v2.7. Figure 5 gives the visualisations for the data used in the original paper and Figure 6 gives the corresponding visualisation for UDPipe 2.0 and UD v2.7.

Figure 7 expands the example plots shown in Figure 1 which only showed extreme cases. This shows LAS versus DUG for every sentence length bin from length 3 to 30. This clearly shows the issue with DUG as discussed in the main body.

All the data used for the analyses presented in this paper can be found in the supplementary material associated with the paper.

	Spearman’s ρ	p-value
DUG LAS	-0.184	0.083
DUG LAS-bcg_{size}	-0.400	0.000
DUG LAS-bcg_{size,Ltest}	0.010	0.926
$\langle L_{\text{Test}} \rangle$ LAS-bcg_{size}	0.465	0.000

Table 7: Correlation of DUG with LAS and then with LAS with the background associated with size and length (L) removed. Isolated row shows correlation of LAS without size background and mean sentence length in test data.

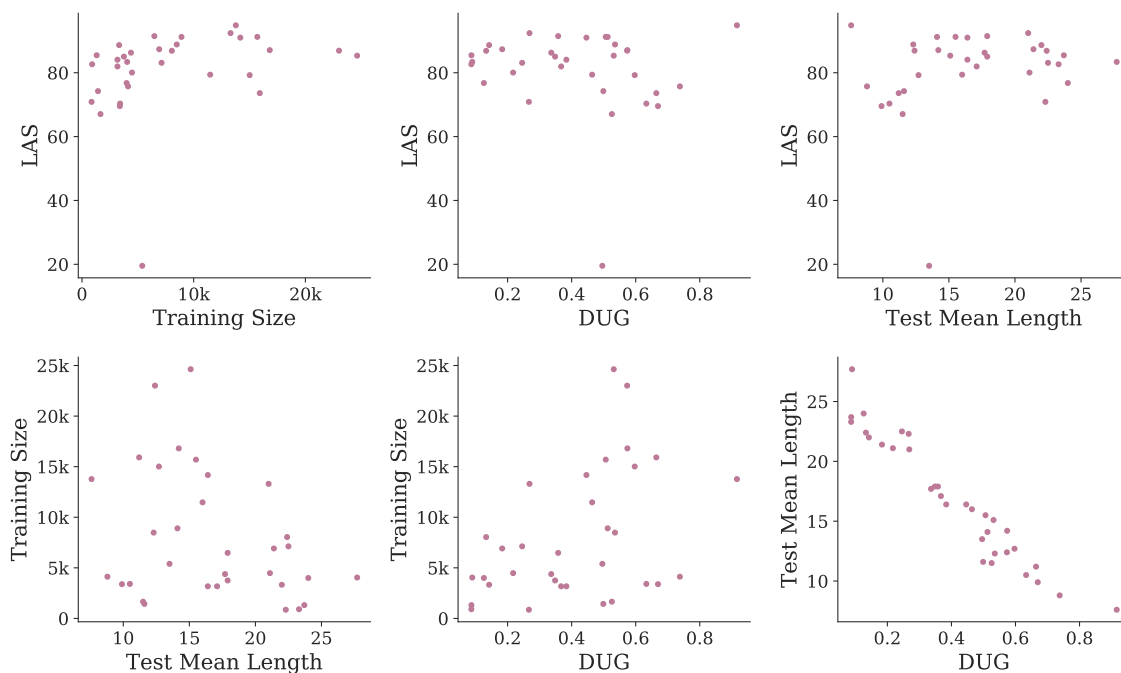


Figure 5: Data from original paper.

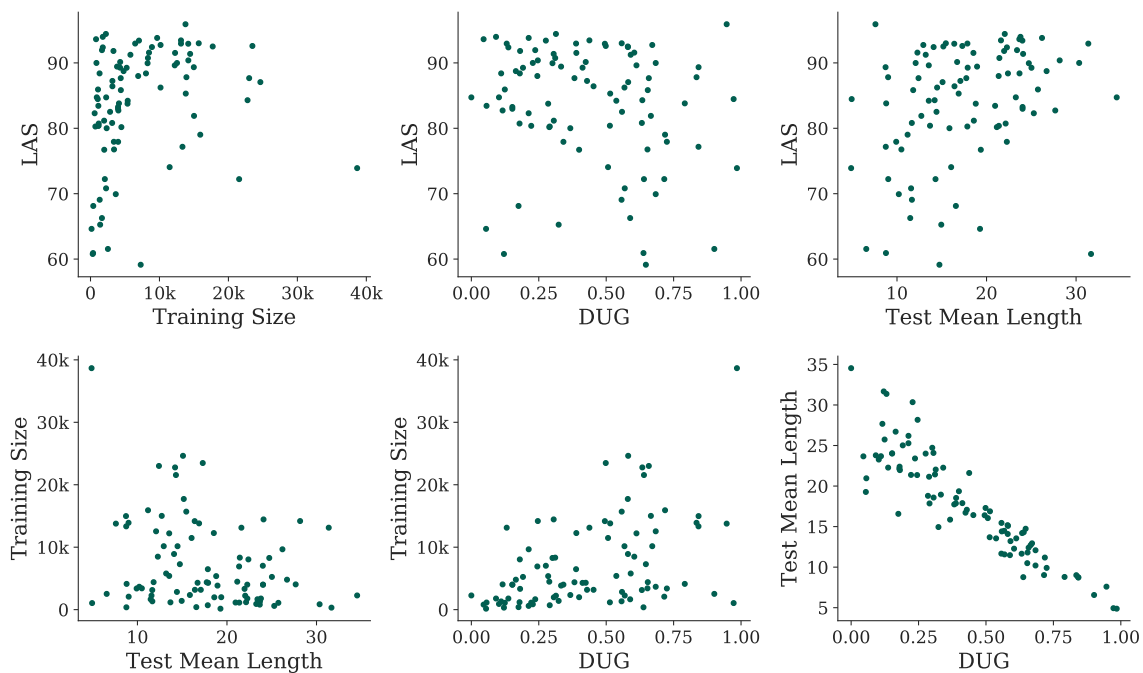


Figure 6: Data for UD Pipe 2.0 and UD v2.7 using DUG.

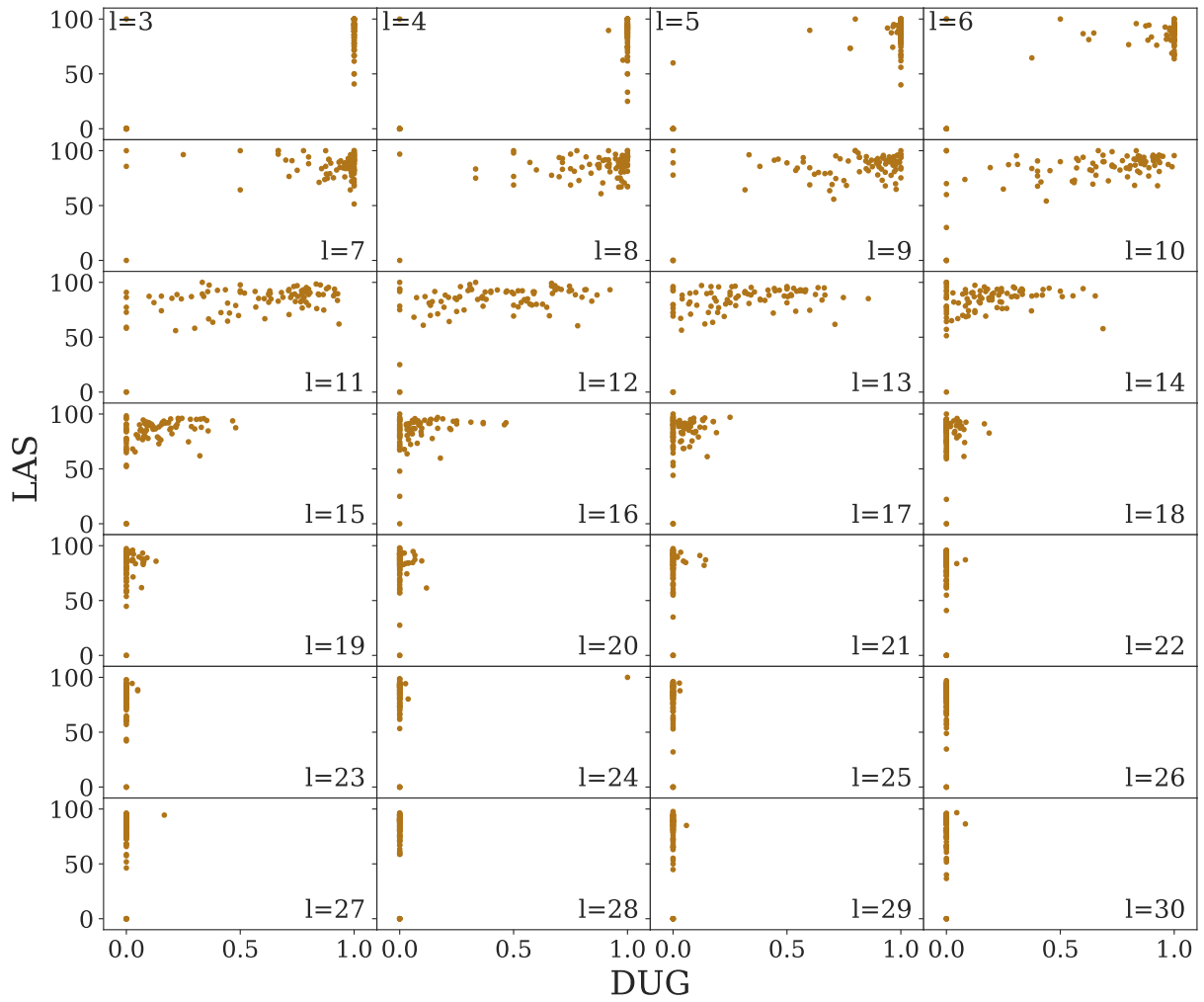


Figure 7: Length-binned analysis. Data for UDPipe 2.0 and UD v2.7 using DUG.