The Impact of Edge Displacement Vaserstein Distance on UD Parsing Performance

Mark Anderson
Universidade da Coruña, CITIC
m.anderson@udc.es

Carlos Gómez-Rodríguez
Universidade da Coruña, CITIC
carlos.gomez@udc.es

We contribute to the discussion on parsing performance in NLP by introducing a measurement that evaluates the differences between the distributions of edge displacement (the directed distance of edges) seen in training and test data. We hypothesize that this measurement will be related to differences observed in parsing performance across treebanks. We motivate this by building upon previous work and then attempt to falsify this hypothesis by using a number of statistical methods. We establish that there is a statistical correlation between this measurement and parsing performance even when controlling for potential covariants. We then use this to establish a sampling technique that gives us an adversarial and complementary split. This gives an idea of the lower and upper bounds of parsing systems for a given treebank in lieu of freshly sampled data. In a broader sense, the methodology presented here can act as a reference for future correlation-based exploratory work in NLP.

1. Introduction

Evaluating the performance of NLP systems is an important task that is often done using a well-established metric or set of metrics. Error analysis often just includes cherry-picking examples that are easy to discuss but don't necessarily give a clear picture of the quality of systems. However, in the context of syntactic parsing, plenty of literature has been written discussing what factors influence parsing performance and it is toward this discussion that this work contributes. We do so by looking at the edge displacement of nodes (the directed distance between the position of the node and its head; see Figure 1) and the corresponding distributions over samples. More specifically, we evaluate the distributions seen in training and test data of treebanks and use the Vaserstein distance to measure the difference between these two distributions. We then compare this with the parsing performance of two different systems that are, broadly speaking, a transition-based and graph-based parser.

Action Editor: Kevin Duh. Submission received: 26 July 2021; revised version received: 11 January 2022; accepted for publication: 25 February 2022.

https://doi.org/10.1162/coli_a_00440

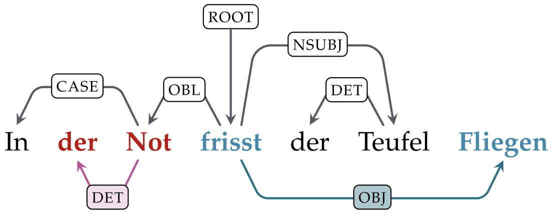

Figure 1
Example tree highlighting dependency displacement for two nodes. **der** at position 2 with its head **Not** at position 3 has a DET edge (in magenta) with a dependency displacement of $2 - 3 = -1$. Similarly, **Fliegen** at position 7 with its head **frisst** at position 4 has an OBJ edge with a dependency displacement of $7 - 4 = 3$. English: *When in need the devil eats flies.*

Hypothesis. We postulate that the differences between the edge dependency displacement distributions of the training and test data of treebanks (as measured by the Vaserstein distance, formally introduced in Section 3.1) are related to the performance of parsers (as defined by the labeled attachment score). We use a number of methods in an attempt to falsify this hypothesis and conclude that based on the data and systems used in this analysis, it cannot be fully refuted. However, the sentence-length binning analysis tempers our complete confidence in this hypothesis.

Utility. We suggest using the observed correlation of Vaserstein distances between edge displacement distributions and parsing performance to guide a sampling method to create adversarial and complementary splits better suited for evaluating parsers.

2. Related Work

In this section we give a brief overview of previous work focused on explaining parsing performance and also focused on dependency distance.

2.1 Analyzing Parsing Performance

An obvious and well-attested predictor of parsing performance is the amount of training data available, which is typically observed to be logarithmically related to parsing performance (Sagae et al. 2008; Falenska and Cetinoğlu 2017; Strzyz, Vilares, and Gómez-Rodríguez 2019; Dehouck, Anderson, and Gómez-Rodríguez 2020). The lengths of sentences have also been observed to impact parsing performance, with longer sentences being harder to parse than shorter sentences (McDonald and Nivre 2011). In a similar vein, others have highlighted the effect that dependency distance has on parsers, namely, that longer dependencies tend to be harder to predict (McDonald and Nivre 2011; Anderson and Gómez-Rodríguez 2020; Falenska, Björkelund, and Kuhn 2020). Edge direction entropy and word order freedom has also been shown to have a meaningful effect (Alicante et al. 2012; Rehbein et al. 2017; Gulordava and Merlo 2015, 2016). This is not consistently observed across all data: Chung, Post, and Gildea (2010) found that for Korean this is not so strongly related to parsing performance as other features of the language such as its pro-drop tendencies. Alicante et al. (2012) only found that it impacted Italian constituency parsing, but not dependency parsing. Part-of-speech bigram perplexity (Berdicevskis et al. 2018), entropy over trees (Corazza, Lavelli, and Satta 2013), the degree of non-projectivity (McDonald and Satta 2007), and morphological complexity (Dehouck and Denis 2018; Cöltekin 2020) have also been presented as explanations or measurements for differences in parsing performance.

Analyses also focus on comparisons between parsing paradigms and algorithms. Transition-based parsers often appear to struggle with longer distance relations more than graph-based parsers (McDonald and Nivre 2011; Falenska, Björkelund, and Kuhn 2020). However, Kulmizev et al. (2019) observed that the use of contextualized word embeddings offset the typical issues associated with transition-based parsers. de Lhoneux, Stymne, and Nivre (2017) investigated the performance of the same transition-based algorithm using a neural network implementation and also a classical implementation, observing the same tendency for performance to decline as dependency distance increased. Anderson and Gómez-Rodríguez (2020) found that the similarity of the inherent displacement distributions of algorithms to the distributions of treebanks was meaningfully correlated with parsing performance when accounting for sentence length for different transition-based algorithms. Beyond this, different frameworks and annotation schemes have been found to perform differently, often related to one or more of the metrics mentioned above (Kübler, Rehbein, and van Genabith 2008; Matsuzaki and Tsujii 2008; Bosco et al. 2010; Mille et al. 2012; Alicante et al. 2012; Pretkalnina and Rituma 2014).

Differences between training and test data have also been evaluated. Zhang and Wang (2009) looked at certain metrics such as the rate of out-of-vocabulary tokens and unseen part-of-speech trigams and observed some correlation between these and parsing performance. However, the main focus in this area is on domain shifts between training and test data. Although this issue is not unique to parsing, there have been extensive results showing that domain shift can result in very steep drops in performance if the domains are very different (Gildea 2001; Bosco et al. 2010; Plank and van Noord 2010; Foster 2010). More recently, Søgaard (2020) proposed the ratio of tree structures in the test data that did not occur in the training data as a predictor of parsing performance, but the results presented were found to be spurious once covariants were accounted for (Anderson, Søgaard, and Gómez-Rodríguez 2021). Here we present a similar analysis but with a measurement that is not so restricted, based on dependency displacement distributions.

2.2 Dependency Distance

Dependency distance is hypothesized to be constrained by working memory restrictions, resulting in distances being minimized (Gibson 2000; Liu, Xu, and Liang 2017). This has been corroborated by numerous corpus-based analyses (Ferrer-i-Cancho 2004; Liu 2008, 2007; Buch-Kromann 2006; Futrell, Mahowald, and Gibson 2015; Temperley and Gildea 2018), although different languages appear to adhere to these restrictions to varying extent (Jiang and Liu 2015; Gildea and Temperley 2010). This relates to NLP parsing because if different languages or treebanks adhere to this constraint more or less than others, it could result in differences in the achievable performance of parsers. Hudson (2017) also highlighted that mean dependency distance varies significantly between treebanks, but added that the direction of dependencies could impact parsing difficulty as well. Different syntactic traits associated with parsing difficulty have been shown to be correlated with an increase in dependency length, for example, free-order languages (Gulordava and Merlo 2015), and with an increase in non-projective dependencies (Ferrer-i-Cancho and Gómez-Rodríguez 2016; Gómez-Rodríguez and Ferrer-i-Cancho 2017).

Gómez-Rodríguez (2017) hypothesized that transition-based parsers perform adequately because they are biased toward short dependencies. This was somewhat corroborated by Eisner and Smith (2010), who improved parser performance by imposing limits on dependency length and using dependency lengths as a feature for their system. It was further substantiated by the work of Anderson and Gómez-Rodríguez (2020) as described in Section 2.1.

The work presented here can be considered an extension of the previous work cited above where we use a method based on edge displacement distributions to compare differences between training and test data to attempt to explain variation in parsing performance across different treebanks.

3. Methodology

In this section we introduce the core principles behind the measurement we focus on in this article and we give the details of the parsing systems and data used in our analysis.

3.1 Edge Displacement Vaserstein Distance

We follow Anderson and Gómez-Rodríguez (2020) and use edge displacement instead of distance as this gives us a measurement that encodes both distance and direction. Fundamentally, it is the signed distance of a node with respect to its head. We alter the definition from Anderson and Gómez-Rodríguez (2020), so that it better resembles the standard definition of physical displacement, that is, the endpoint minus the starting point:

$$s_{\text{edge}} = x_{\text{node}} - x_{\text{head}} \tag{1}$$

Then for a given treebank the edge displacement for each node is measured, excluding the root node and its displacement with respect to the dummy root as position 0. The range $[-30, 30]$ is used for the distributions so that the measurement isn't impacted by potential unreliable long tails. This range covers 99.40% of the edges in UD v2.5 and 99.38% in UD v2.6. The distribution of edge displacements is then normalized such that it takes the form of a probability distribution. In this way, a probability distribution over displacements is obtained for the training treebank and test treebank for each dataset. We then use these two probability distributions to calculate the Vaserstein distance (Vaserstein 1969), thus obtaining the edge displacement Vaserstein distance (EDV) for a given dataset. The Vaserstein distance (technically the Vaserstein-1 distance) is defined as follows (Vaserstein 1969):

$$\ell(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu, \nu)} \int |x - y| d\gamma(x, y) \tag{2}$$

where μ and ν are probability distributions of two random variables (in our case, the variables will correspond to dependency displacements), x and y are points in the x-axis of these probability distributions (i.e., concrete values of each of the variables), $|x - y|$ is the distance between two such values, and the infimum is with respect to γ, a coupling from Γ which is the set of all joint distributions whose marginals are μ and ν.

A more grounded interpretation of the Vaserstein distance is that it gives a measurement of how much *mass* needs to be moved from each x to each y so that μ is transformed into ν. As such, this metric is also known as earth mover's distance in computing

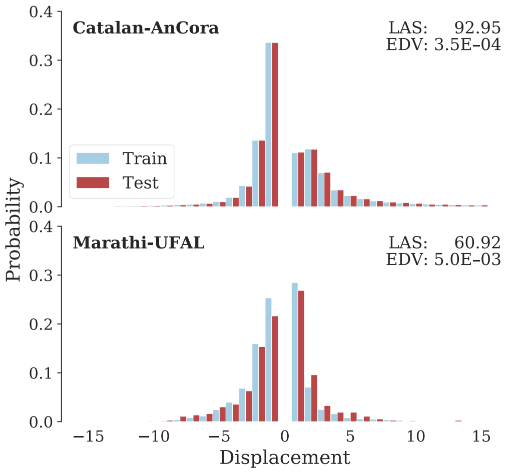

Figure 2
Example displacement distributions of the training and test data for Catalan-AnCora (top) and Marathi-UFAL (bottom), which exhibit the smallest and largest measured EDV values in UD v2.6. While both EDV values are small, there is an order of magnitude difference between them. LAS is shown for UDPipe 2.0.

science. Ultimately, it gives a measurement of how different two distributions are, with larger values indicating a greater divergence and values approaching zero indicating similar distributions.

Example. Distributions are shown for two treebanks from the Universal Dependency (UD) v2.6 treebanks in Figure 2. As can be seen, Catalan-AnCora has very similar distributions for its training and test data, which is reflected in a small EDV of 3×10^{-4}. Marathi-UFAL is also shown, where differences between the two sets can be clearly seen despite the distributions following similar trends. This still results in a small EDV of 5×10^{-3}, but it is an order of magnitude greater than that observed for Catalan-AnCora. These two treebanks show the highest EDV (Marathi-UFAL) and the lowest (Catalan-AnCora), and so show the range of EDV values observed in the data (the mean EDV observed in UD v2.6 is $1.40(0.85) \times 10^{-3}$, and $1.35(0.87) \times 10^{-3}$ for UD v2.5). Despite the values of EDV both being fairly small, there is a large difference in performance seen for these two treebanks, with Catalan-AnCora achieving a labeled attachment score (LAS) of 92.95 when using UDPipe 2.0, and Marathi-UFAL only achieving 60.92. There are clearly other contributing factors relating to the difference in performance between these two treebanks (not least training data size, as Marathi-UFAL only has 373 training instances whereas Catalan-AnCora has 13,123), which we have discussed above and that we take into consideration in our analysis discussed below.

3.2 Parser Systems

We used two neural-based parsers: version 1.2.1-devel (1.2) of UDPipe, and version 2.0 (Straka and Straková 2017; Straka 2018). For UDPipe 1.2 we use models 2.5[1] and for

[1] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3131.

UDPipe 2.0 we use models 2.6.[2] We opted to use these systems as the models have been optimized for their respective UD treebank and UDPipe 1.2 is a transition-based system whereas UDPipe 2.0 is a graph-based system, thus allowing us to evaluate EDV for different parser systems. Furthermore, UDPipe 1.2 came 8th out of 33 at the CoNLL 2017 shared task and was used as the baseline model for comparison of systems submitted to the CoNLL 2018 shared task, where it came 18th out of 26 with respect to average LAS. For its part, UDPipe 2.0 was one of the top performing parsers of the 2018 shared task, tied for the 3rd place (Zeman et al. 2017, 2018). An earlier version of UDPipe 2.0 was also one of the leading systems at the SIGMORPHON 2019 shared task, and the winner of EvaLatin 2020 (McCarthy et al. 2019; Sprugnoli et al. 2020).

Both systems include tokenization and sentence segmentation capabilities, but we fed gold tokenized data to the systems, as we are interested in the impact of EDV on parsing specifically and not how it relates to these preliminary tasks. When running the systems, we opted to run the taggers when parsing so as to use the systems close to how they were intended to be used, even though we are not interested in the tagging performance (of UPOS and mfeats). This results in using predicted tags at runtime.

UDPipe 1.2 is a basic feed-forward neural transition-based parser which uses a simple feature function as input for each timestep (Chen and Manning 2014; Straka et al. 2015). We used models 2.5 which were pre-trained on UD v2.5 treebanks, resulting in 94 parsers on separate treebanks. Each model is optimized for each treebank, which includes the type of algorithm and oracle used. Details of the system can be found in Straka et al. (2015) and Straka, Hajič, and Straková (2016).

UDPipe 2.0 is based on the graph-based biaffine parser of Dozat and Manning (2017) where the hidden representations of tokens from BiLSTM layers are mapped into two separate perceptron layers, considered representations of the tokens as a head and as a dependent, which are combined using a biaffine attention mechanism, resulting in a probability distribution over all other tokens in a sentence indicating the probability that any given token is its head. A well-formed tree is then enforced using the Chu–Liu/Edmonds' algorithm (Chu and Liu 1965; Edmonds 1967). We could not run Czech-PDT, Hindi-HDTB, German-HDT, and Russian-SynTagRus, as the Web site had issues with large files, so we ended up with results from 90 models. Note that although the treebanks used for UDPipe 1.2 and 2.0 are very similar, they are not exactly the same. There are a few differences in the actual treebanks included and there are also differences within given treebanks between iterations of UD releases.

Data. We used UD treebanks for our analysis (as such, we lay no claim to any results that span different frameworks). We used the sets of treebanks that correspond to the parser models we used for each system, namely, UD v2.6 with UDPipe 2.0 and UD v2.5 for UDPipe 1.2. We also used UD v2.7 to extend our analysis beyond the pretrained model for evaluation of the linear regression model using unseen data. We picked treebanks that had no UDPipe 1.2 model but contained both training and test data and that contained at least 100 sentences in the training data. We also used UD v2.7 for a proof of concept for using EDV to guide sampling for a more robust evaluation procedure for parsers. This resulted in 94 treebanks for UDPipe 1.2, 90 treebanks for UDPipe 2.0, 11 treebanks for evaluating the UDPipe 1.2 linear model, and 105 treebanks for the sampling work.

[2] `https://lindat.mff.cuni.cz/services/udpipe/`.

3.3 Statistical Methods

The statistical analysis was undertaken using the Pingouin Python library version 0.3.8, except the partial coefficients (§4.2.3) were calculated using version 0.4.0, which corrected errors associated with these (Vallat 2018).

Correlation Coefficients. We evaluate the impact variables have on parsing performance by measuring their correlation coefficients with respect to LAS. We use a non-parametric correlation coefficient in the form of Spearman's ρ, which measures the correlation between variables and assesses the monotonic relationship between them. We do not use Pearson's r, as the data being analyzed do not strictly adhere to bivariate probability distributions and the sample sizes are small enough that this can affect the measurement's sensitivity. Further, Pearson's r is less robust with respect to outliers. For each coefficient, we report the correlations and the corresponding p-value. For the main correlation results, we include the upper and lower bounds of the 95% confidence interval, the coefficient squared (a measure of the proportion of explained variance), the adjusted coefficient (which somewhat tempers the coefficient's bias), and the power of the analysis. For p-values, we report the exact value unless the value is less than 0.001, following common practice (American Psychological Association 2010).

Partial Correlations. We make use of partial correlations to evaluate the impact of covariants. This allows us to remove the impact of variables that are correlated with the control variable and the target variable, so as to avoid situations where a measurement seemingly explains X variance in the data but in reality it is merely a measurement of one or more basic variables.

Background Removal. Here we take a standard method found in physics used to remove known background functions from data, for example, removing the spectra associated with amorphous radiators from those associated with lattice-structure radiators to obtain enhanced spectra that is without noise (Timm 1969). Here we consider the variations associated with covariants as similar background data to be removed, so as to observe if there is any variation associated with EDV. Similar to partial correlations, removing the background signal of a potential covariant allows us to visually evaluate the specific impact a variable of interest has on the target variable. This involves fitting the control data and the target (e.g., the size of training data and LAS) and then dividing the target variable by the predicted values from this fit. This *normalized* data is then used to fit a second potential covariant which too is used to divide the normalized target variable values. This can be repeated for any number of covariants. Ultimately, a normalized version of the target variable is left and the control target of interest (e.g., EDV) is evaluated against these values and if a trend is still observed, it is evidence that this variable has an impact on the target variable even with the variance associated with these covariants removed. This technique ultimately acts as a way of tempering correlations we calculate and gives us a means of disentangling contributions that might not be caught by partial correlation calculations.

Linear Regression. The preceding methods allow us to hone in on the impact of a given variable, but with linear regression we can fit models to the data with more than one variable. This allows us to evaluate the impact certain variables have when used with other covariants. For linear regression models we report the adjusted R^2 (the square of the residuals) as a measurement of the proportion of explained variance, which it equals

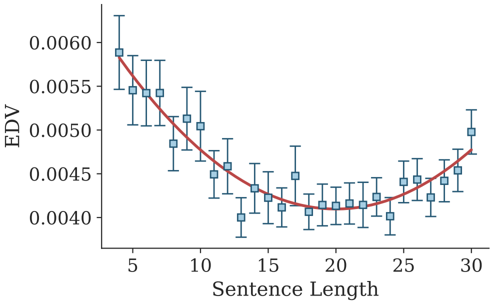

Figure 3
EDV between sub-samples of the training and test data binned by sentence length for UD v2.6 (111 treebanks).

when the residual mean is normalized so as to equal zero (as is the case in this analysis). In addition, we report the relative importance of each variable and the corresponding p-values (Sen et al. 1981; Groemping 2006).

Sentence Length Binning. Ferrer-i-Cancho and Liu (2014) highlighted the impact mixing sentence lengths can have on treebanks analyses and Anderson and Gómez-Rodríguez (2020) observed sentence-length dependencies when evaluating edge displacement distributions of treebanks and the inherent distributions of transition-based parsers. Considering this potential impact, we also undertake a sentence-length binned analysis. This simply entails constructing samples of each treebank based on the length of the sentences. We take bins ranging from 3 tokens to 30 tokens, as any shorter and the EDV has little meaning (i.e., with 2 tokens, there can only be one edge which can either be –1 or 1) and any longer and the number of instances in a given bin for a given treebank is too small to obtain a meaningful measurement. Note that parsers were trained on the full data and the binning procedure is undertaken solely at the analysis stage. Figure 3 shows the EDV calculated between training and test data for each sentence length bin for UD v2.6 (the corresponding data for UD v2.5 is shown in Figure A.1 in Appendix A). It is clear that EDV does vary based on sentence length, but it remains to be seen whether that variation has an impact on parsing performance.

Variables Assessed. Beyond assessing EDV and how it correlates to parsing performance (as given by LAS) we look at a number of variables that are potential covariants. First we look at the size of the training data (measured both in tokens and sentences), which as described above has been shown to correlate to parsing performance and could feasibly impact EDV measurements. That is, larger treebanks allow for a more *accurate* representation of a language's true underlying distributions of edge displacements so deviations with respect to the test data could be minimized, and vice versa: If the sample is too small, it could be some random sample at the fringes of what would be a standard distribution for a given language. Similarly, we also consider the number of tokens and sentences in the test data. We also look at the mean sentence length of the test data, $\langle L_{\text{test}} \rangle$, as this theoretically puts a limit on the potential distribution of edge displacements and has been observed to impact parsing performance (i.e., longer sentences are harder to parse than shorter ones). For the sake of completeness, we also look at the mean length of the training data, $\langle L_{\text{train}} \rangle$. Finally, we look at the Vaserstein distance between the training and test distributions of sentence lengths (SLV) because it is feasible that EDV merely vaguely measures differences with respect to sentence length.

4. Analysis and Results

In this section we describe the analysis in detail and discuss the results we obtained.

4.1 Evaluating Normality

Here we justify the use of Spearman's ρ for the following analysis. Figure 4 shows the distribution of the variables of interest in our analysis (as described in Section 3.3) for UD v2.6 (the corresponding distributions for UD v2.5 are shown in Figure A.2 in Appendix A). Visually, it is clear that only $\langle L_{\text{test}} \rangle$ could be sampled from a normal distribution.

To thoroughly evaluate the variables for normality, we use the Shapiro–Wilk test (Shapiro and Wilk 1965), as it is a higher power test compared with the alternatives, making it the most suitable for our fairly small sample size (Yap and Sim 2011). The values from the tests (W) and the corresponding p-values (where the null hypothesis is that the sample *is* from a normal distribution) are shown in Table 1 for both UD v2.5 (top) and UD v2.6 (bottom). A smaller W indicates that a sample is not drawn from a normal distribution, but the more informative metric here is the p-value (as W is nonlinear and difficult to interpret). Basically, larger p-values mean we cannot reject the null hypothesis that the sample is drawn from a normal distribution. Only $\langle L_{\text{test}} \rangle$ has a large p-value and does so for both datasets (0.121 for UD v2.5 and 0.402 for UD v2.6). The leftmost column of Table 1 shows the result of the test based on the

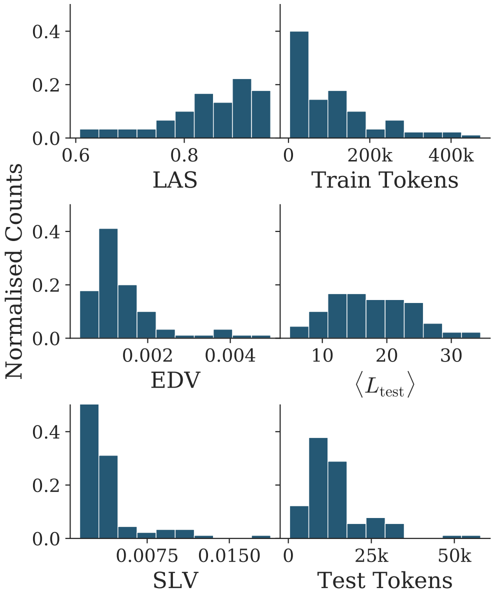

Figure 4
Distributions of the variables of interest in UD v2.6 (90 treebanks) in order to evaluate whether they are sampled from normal distributions.

Table 1
Shapiro–Wilk tests to evaluate if samples are drawn from normal distributions for UD v2.5 (top) and UD v2.6 (bottom). Only the $\langle L_{\text{test}} \rangle$ test has values for which the null hypothesis (i.e., normal distribution) cannot be rejected under any reasonable thresholds.

Variable	W	p-value	Normal
LAS	0.920	<0.001	False
Train Tokens	0.418	<0.001	False
EDV	0.785	<0.001	False
$\langle L_{\text{test}} \rangle$	0.978	0.121	**True**
SLV	0.686	<0.001	False
Test Tokens	0.350	<0.001	False
LAS	0.894	<0.001	False
Train Tokens	0.851	<0.001	False
EDV	0.761	<0.001	False
$\langle L_{\text{test}} \rangle$	0.985	0.402	**True**
SLV	0.665	<0.001	False
Test Tokens	0.825	<0.001	False

ever arbitrary distinction of significance, that is, p-value < 0.05. We are not particularly interested if one variable is or is not normally distributed; the important result here is that most variables including the control variable of interest (EDV) and the target variable (LAS) quite definitively do not follow normal distributions. This, along with the other considerations mentioned in Section 3.3, thoroughly justify the use of Spearman's ρ. Further, it is useful that this coefficient doesn't specifically evaluate the linearity of relationships because not all variables assessed here are linearly related to parsing difficulty, but *are* monotonically related.

4.2 Correlation Coefficients

Here we evaluate basic coefficients between the control variables and LAS and also between the potential covariants and EDV.

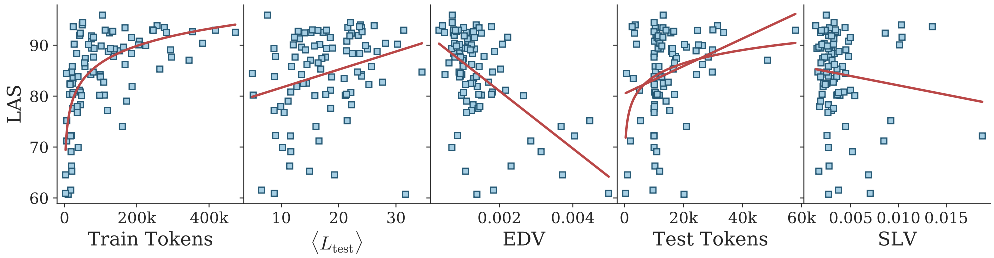

Figure 5
Visualization of LAS (for UDPipe 2.0 and UD v2.6) with respect to variables of interest with fits shown in red to highlight whether the data appears correlated or not.

4.2.1 Basic Coefficients. Figure 5 shows LAS against the control variables of interest for UDPipe 2.0 (the corresponding visualization for UDPipe 1.2 is shown in Figure A.4 in Appendix A). In the first subplot, it is fairly clear that LAS increases logarithmically with respect to the number of tokens in the training data, which corroborates the findings discussed above in Section 2. It appears that the number of tokens in the test data is not associated with parsing performance for UDPipe 2.0, however, there is a potentially logarithmic relationship seen for UDPipe 1.2, but that could easily be down to a few serendipitously placed outliers. $\langle L_{\text{test}} \rangle$ is loosely linearly related to LAS, but EDV seems like it is more strongly linearly related. SLV doesn't seem to be related to LAS, but there are a few clusters which upset the fitting procedure that should not affect the calculation of the corresponding Spearman ρ for this relation. Note that we do not visualize all variables for the sake of space and to avoid redundancy, that is, the number of training tokens is more strongly correlated to parsing performance than the number of training sentences (as seen in Table 2).

Table 2 shows the corresponding Spearman ρ values for the data shown in figures 5 and A.4 and the remaining variables mentioned above in Section 3.3, that is, control variables related to LAS. First, we want to note that measuring data in tokens rather than sentences results in stronger correlations for both test and training and for both parsers (with the number of test instances not even being correlated to LAS for UDPipe 2.0). Based on this, we use the number of tokens in the training and test data from this point forward. Also, the number of training tokens is the variable most strongly correlated with parsing performance, but the next strongest for both systems (excluding the number of training sentences) is actually EDV. SLV is not correlated at all for UDPipe 2.0 and only weakly so for UDPipe 1.2, with a p-value higher than any arbitrary threshold of significance.

Next we investigate how the variables most strongly correlated to LAS correlate with one another, that is, we check for potential covariants. Figure 6 shows how pertinent variables relate to EDV. Clearly, the number of tokens in the training data and the

Table 2
Spearman's ρ for correlations between variables of interest and LAS.

Parser	Variable	ρ	p-value
UDPipe 1.2	Train Tokens	0.660	<0.001
	Train Trees	0.535	<0.001
	$\langle L_{\text{train}} \rangle$	0.376	<0.001
	Test Tokens	0.433	<0.001
	Test Trees	0.208	0.045
	$\langle L_{\text{test}} \rangle$	0.351	0.001
	SLV	−0.191	0.065
	EDV	−0.492	<0.001
UDPipe 2.0	Train Tokens	0.605	<0.001
	Train Trees	0.467	<0.001
	$\langle L_{\text{train}} \rangle$	0.323	0.002
	Test Tokens	0.309	0.003
	Test Trees	0.073	0.496
	$\langle L_{\text{test}} \rangle$	0.309	0.003
	SLV	−0.086	0.422
	EDV	−0.466	<0.001

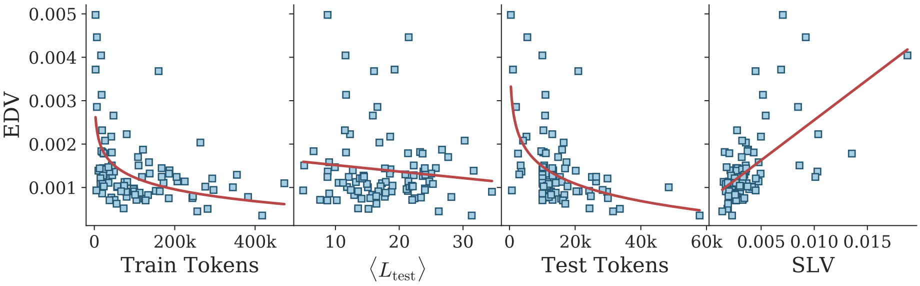

Figure 6
Visualization of EDV (for UD v2.6) with respect to variables of interest with fits shown in red to highlight whether the data appears correlated or not.

test data are strongly related and, as one would expect, SLV looks related (confirmed by the actual correlation coefficient of 0.549 with a p-value less than 0.001, as seen in Table 3). However, as SLV is not correlated to parsing performance, it is not necessary to consider it when evaluating EDV with respect to LAS. It seems like $\langle L_{\text{test}} \rangle$ is not clearly related to EDV despite our expectations that it would be.

The corresponding correlations are found in Table 3 alongside correlations between other variables as well. The correlations clearly corroborate the trends observed in Figure 6. $\langle L_{\text{train}} \rangle$ is not shown in Figure 6 but it behaves similarly to $\langle L_{\text{test}} \rangle$, closely echoing the measured correlations between $\langle L_{\text{test}} \rangle$ and EDV for both systems. We also show the correlation between the number of training tokens and test tokens because typically the amount of data for both are linked (i.e., it is not particularly common for a treebank to have a huge training set but a tiny test set, although the opposite does occur, e.g. Kazakh-KTB). For both sets of data the correlations are high (0.772 for UD v2.5 and 0.659 for UD v2.6) both with p-values below 0.001. We assume, therefore, that these measurements loosely capture the same aspect of treebanks and use the number

Table 3
Spearman's ρ for different pairs of variables.

Parser	Variables	ρ	p-value
UDPipe 1.2	Train Tokens — EDV	−0.480	<0.001
	$\langle L_{\text{test}} \rangle$ — EDV	−0.080	0.443
	$\langle L_{\text{train}} \rangle$ — EDV	−0.089	0.393
	Test Tokens — EDV	−0.523	<0.001
	SLV — EDV	0.617	<0.001
	Test — Train (Tokens)	0.772	<0.001
	$\langle L_{\text{test}} \rangle$ — Train Tokens	0.149	0.153
UDPipe 2.0	Train Tokens — EDV	−0.424	<0.001
	$\langle L_{\text{test}} \rangle$ — EDV	−0.025	0.817
	$\langle L_{\text{train}} \rangle$ — EDV	−0.023	0.833
	Test Tokens — EDV	−0.446	<0.001
	SLV — EDV	0.549	<0.001
	Test — Train (Tokens)	0.659	<0.001
	$\langle L_{\text{test}} \rangle$ — Train Tokens	0.096	0.370

of training tokens as the best option: It is more strongly correlated to LAS by a large amount and is similarly correlated to EDV if slightly less so than the number of test tokens. We further justify this choice in Section 4.2.2. Lastly, we show the correlation of $\langle L_{\text{test}} \rangle$ and the number of training tokens as it has been noted that smaller treebanks (especially very low-resource treebanks) not only have less training instances but also sentences tend to be shorter (Dehouck and Gómez-Rodríguez 2020). However, we don't find any correlation in these datasets, presumably because this issue is not prevalent once a certain threshold of data size is reached.

4.2.2 Background Removal. As described above, we removed the background signal associated with other variables to evaluate the independent relationship of certain variables. First, we evaluated whether the number of test tokens actually captured a different aspect of the treebanks with respect to parsing performance. Figure 7 shows this process for UDPipe 1.2, where the first plot shows LAS against the number of training tokens and the second plot shows the normalized LAS (LAS / fit from first plot) against test tokens.

We show this process for UDPipe 1.2 rather than 2.0, which we have used for the visual representations in the main body thus far (the corresponding plot for UDPipe 2.0 is shown in Figure A.3 in Appendix A), as the visual relationship observed for UDPipe 1.2 between the number of test tokens and LAS was much more convincing than for UDPipe 2.0 and the correlation reported in Table 3 was higher for UDPipe 1.2. It is clear that once we remove the signal associated with the number of training tokens, the signal associated with the number of test tokens disappears. This is backed up by the correlations observed for the number of test tokens and LAS (0.433, p-value < 0.001) disappearing when comparing the number of training tokens to the normalized LAS with a correlation of -0.123 (p-value $= 0.236$).

We note here that when looking at the partial coefficient for the number of test tokens for UDPipe 1.2 when using the number of training tokens as a covariant, we obtain a coefficient of -0.325 (p-value $= 0.001$), which is not particularly meaningful and highlights the fragility of correlation coefficients. In fact, the reversal of the

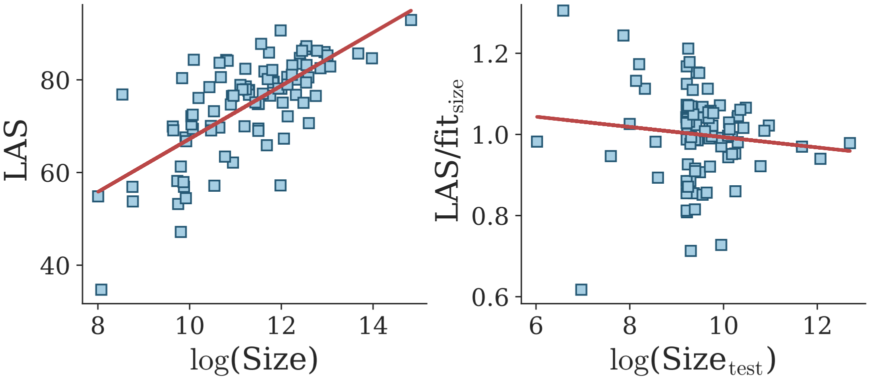

Figure 7
Background removing method used to evaluate whether the number of test tokens carries additional information with respect to the number of training tokens for UDPipe 1.2 and UD v2.5. Correlation between the number of test tokens and LAS is 0.433 (p-value < 0.001) and that between the number of test tokens and the normalized LAS (right plot) is -0.123 (p-value $= 0.236$).

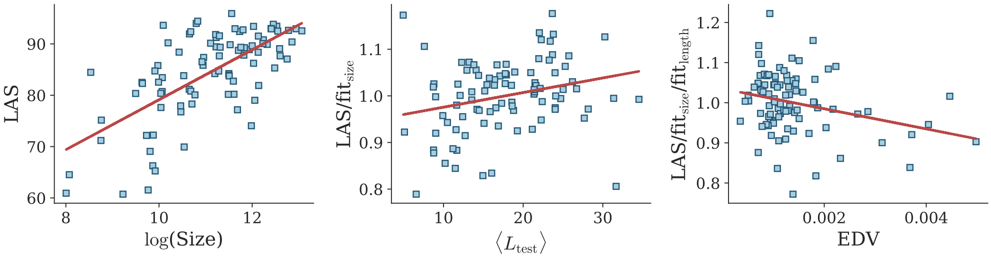

Figure 8
Background removal method to evaluate whether a correlation is observed between EDV and LAS (for UDPipe 2.0 and UD v2.6) after removing the variation associated with the training test size and $\langle L_{\text{test}} \rangle$. The correlation between EDV and LAS is -0.466 (p-value < 0.001), the correlation between EDV and the LAS normalized by the variance associated with number of tokens in training data is -0.222 (p-value $= 0.036$), and the correlation for the fully normalized LAS (removing the variance associated with $\langle L_{\text{test}} \rangle$) is -0.283 (p-value $= 0.007$).

sign is indicative of multicollinearity, exactly what we anticipated these variables to be (Farrar and Glauber 1967). For UDPipe 2.0 the same partial correlation is -0.045 (p-value = 0.671) and so it is even clearer for this system.

We next use this technique to evaluate the relationship observed between EDV and LAS. In Figure 8 we show the fit of LAS against the number of training tokens (leftmost plot), and then the first normalized LAS against $\langle L_{\text{test}} \rangle$ (middle plot), and the final normalized LAS against EDV (rightmost plot) for UDPipe 2.0 (Figure A.6 in Appendix A shows the equivalent analysis for UDPipe 1.2). We opted to include $\langle L_{\text{test}} \rangle$ even though no correlation was observed between $\langle L_{\text{test}} \rangle$ and EDV because theoretically it could impact the measurement of EDV, and if the coefficients failed to capture this, it could still impact the final analysis. However, removing the signal associated with it and the number of training tokens still results in a clear linear relationship between EDV and LAS (correlation of -0.283 with p-value $= 0.007$). The correlation is much diminished compared to the original coefficient measured for EDV of -0.466 (Table 2), but it is still meaningful. The results are echoed in the analysis for UDPipe 1.2 with a correlation of -0.249 (p-value $= 0.015$) between EDV and the final normalized LAS compared to -0.492 for the original measured coefficient (Table 2).

4.2.3 Partial Coefficients. This ultimately leads us to evaluating EDV with respect to LAS using partial coefficients. The main covariant of interest is the number of tokens in the training data, which is not only the most strongly correlated variable with respect to LAS (Table 2) but also the second most strongly correlated variable with respect to EDV (Table 3). We also include $\langle L_{\text{test}} \rangle$ despite measuring no correlation with it and EDV because of the apparent impact it had in the background subtraction analysis (Section 4.2.2). In Table 4, we show the full measurement of the partial coefficients for EDV with respect to LAS for UDPipe 1.2 and 2.0 with no covariants (i.e., the standard coefficient), with the number of training tokens as the sole covariant, and with both the training tokens and $\langle L_{\text{test}} \rangle$ as covariants. As expected, when evaluating the correlation with the number of training tokens as a covariant we observe the biggest change in the measured coefficient. For UDPipe 1.2 it drops from -0.492 to -0.265 and for UDPipe 2.0 it drops from -0.466 to -0.290. We also note that despite not being correlated based on the calculated coefficients between $\langle L_{\text{test}} \rangle$ and EDV, we still checked its impact. There

Table 4
Partial coefficients (except for rows with None in Covariant(s) column) for EDV with respect to LAS for UDPipe 1.2 and UD v2.5 (top) and for UDPipe 2.0 and UD v2.6 (bottom). Shown is the coefficient itself (ρ), the 95% confidence interval (CI95%), ρ^2 as an indication of the proportion of explained variance, the adjusted ρ^2 (Adj. ρ^2) as a less biased version of ρ^2, the corresponding p-values, and the achieved power of the test (power).

Parser	Covariant(s)	ρ	CI95%	ρ^2	Adj. ρ^2	p-value	power
UDPipe 1.2	None	−0.492	[−0.63 −0.32]	0.242	0.234	<0.001	0.999
	Train Tokens	−0.265	[−0.44 −0.06]	0.070	0.050	0.010	0.735
	Train Tokens, $\langle L_{\text{test}} \rangle$	−0.278	[−0.46 −0.08]	0.077	0.047	0.007	0.773
UDPipe 2.0	None	−0.466	[−0.61 −0.29]	0.217	0.208	<0.001	0.997
	Train Tokens	−0.290	[−0.47 −0.09]	0.084	0.063	0.006	0.796
	Train Tokens, $\langle L_{\text{test}} \rangle$	−0.312	[−0.49 −0.11]	0.097	0.066	0.003	0.849

is a small increase in the partial coefficients here signaling that $\langle L_{\text{test}} \rangle$ is not a covariant of EDV with respect to LAS. This partial correlation coefficient results in an adjusted ρ^2 of 0.047 for UDPipe 1.2 and 0.066 for UDPipe 2.0, which gives a less biased indication of the proportion of explained variance associated with EDV (5% for UDPipe 1.2 and 7% for UDPipe 2.0). Only including the number training tokens as a covariant results in an adjusted ρ^2 of 0.050 for UDPipe 1.2 and 0.063 for UDPipe 2.0 (5% and 6%, respectively). Therefore, in this setting, we can say that EDV is correlated with a non-trivial amount of the differences observed in parsing performance across treebanks.[3]

4.3 Multilinear Regression

We then evaluated the impact EDV has in a multilinear regressive fit of the data for both systems. The results are shown in Table 5. We start by simply fitting a model using the log of the number of training tokens and for both systems we obtain a fit that has reasonably large adjusted R^2 (0.475 and 0.434 for UDPipe 1.2 and 2.0, respectively). We also use $\langle L_{\text{test}} \rangle$ based on the results from Sections 4.2.2 and 4.2.3 and see that the adjusted R^2 for the model using this and the log of training token size is slightly higher than only using the training tokens (about 0.03 for both systems). Using training tokens with EDV, however, results in a larger increase of 0.09 for UDPipe 1.2 and 0.06 for UDPipe 2.0. We also observe an increase when using EDV in addition to the other two variables, which results in the largest adjusted R^2 of 0.589 and 0.522 for UDPipe 1.2 and 2.0, respectively.

It is necessary to highlight that despite reporting the adjusted R^2, it is still a biased indication of the proportion of explained variance of a model. However, it is still indicative of the quality of the model, but more importantly it allows us to evaluate the impact of EDV. We also report the relative importance percentages in Table 5, which show that EDV roughly carries 40% of the importance in the models it is used in for UDPipe 1.2 and about 35% for UDPipe 2.0.

[3] However, interpreting correlations is somewhat subjective. Others might see these values and surmise that EDV is less informative than the training data size on its own and only adds a small amount of additional explanation of the observed variation in parsing performance. We have attempted to report the statistics in a way that readers can come to their own conclusions while also offering our personal interpretations.

Table 5
Statistics associated with linear regression models using combinations of log size, EDV, and $\langle L_{\text{test}} \rangle$ as predictors. We report the adjusted R^2 scores for linear regression fits as a less biased indication of the proportion of explained variance and report the percentage of relative importance of each predictor along with the corresponding p-values.

Parser	Variables	Adj. R^2	Relative Importance	p-values
UDPipe 1.2	logTrain Tokens	0.475	100.0	<0.001
	logTrain Tokens, $\langle L_{\text{test}} \rangle$	0.503	87.8, 12.2	<0.001, 0.015
	logTrain Tokens, EDV	0.567	55.7, 44.3	<0.001, <0.001
	logTrain Tokens, $\langle L_{\text{test}} \rangle$, EDV	0.589	50.8, 8.6, 40.6	<0.001, 0.018, <0.001
UDPipe 2.0	logTrain Tokens	0.434	100.0	<0.001
	logTrain Tokens, $\langle L_{\text{test}} \rangle$	0.468	88.3, 11.7	<0.001, 0.012
	logTrain Tokens, EDV	0.494	61.4, 38.6	<0.001, 0.001
	logTrain Tokens, $\langle L_{\text{test}} \rangle$, EDV	0.522	56.0, 9.1, 35.0	<0.001, 0.015, 0.001

4.3.1 Testing the Model with UDPipe 1.2. As there exists a more up-to-date version of UD that contains more treebanks not used in the systems we have evaluated, we can use these new treebanks to evaluate the linear model from Section 4.3. We select the new treebanks based solely on two criteria: that the treebanks have at least 100 training sentences (as very small treebanks tend to be very volatile with respect to performance) and that they contain pre-existing training and test sets (and potentially a development set). This resulted in 11 new treebanks. Note, Latin-LLCT fit these criteria but we opted not to use it, as it contains the same sentence 356 times across the training, development, and test data.

We trained models using UDPipe 1.2 with the general settings. This means these data points are slightly different from those used to develop the linear regression model that were all optimized for each treebank based on the algorithm and oracle used. We ran the evaluation the same as described in Section 3.2. We did not train models for UDPipe 2.0 as the parser is not publicly available. We then compared the LAS we obtained from these parsers and the values predicted by the linear regression model using all 3 variables, as discussed in Section 4.3. The comparisons are shown in Figure 9, where the predicted values are not outlandishly different for most treebanks except for those that obtained fairly low LAS. While we have not set out to develop a predictive model, this is still useful as a sanity check (if the predictions had been wildly inaccurate across the board, then one would have to question not only the linear model but the calculated coefficients).

4.4 Sentence Length Binning

Here we turn to our sentence length binning analysis. As shown above in Figure 3 (and Figure A.1 in Appendix A), EDV does show an expected dependency on sentence length. We also would like to highlight that this dependency is hardly unique to this situation, but consideration of this is almost completely lacking in NLP. Figure 10 shows the partial correlation coefficients and the corresponding p-values for each sentence length bin we evaluated in this analysis (sentence lengths of 3 to 30) for both parsers. Note that we only used the number of tokens in the training data as a covariant because for each bin $\langle L_{\text{test}} \rangle$ is constant across each treebank by design.

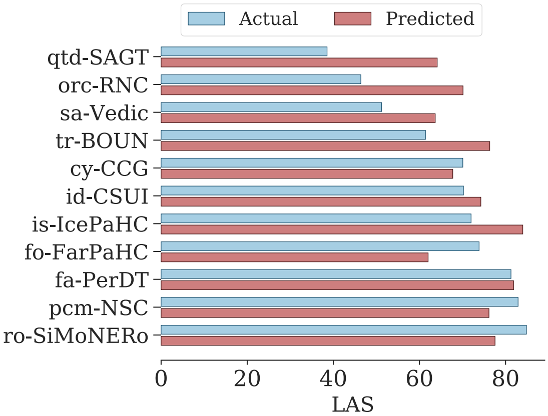

Figure 9
Comparison of performance of new UDPipe 1.2 models for treebanks not covered in current UDPipe 1.2 models that appear in UD v2.7 with predictions from linear model from Section 4.3 using the log of the number of training tokens, $\langle L_{\text{test}} \rangle$, and EDV as predictors. The mean absolute error is 11.05.

A clear trend can be observed where the magnitude of the correlations increases as a function of sentence length. However, most correlations don't have a particularly low p-value, with the largest sentence-length bin being the exception. We offer visualization of the corresponding scatter plots for each bin in figures A.7 and A.8 in Appendix A for UDPipe 1.2 and 2.0, respectively. From these plots, it appears that there are some linear relations that echo the correlation coefficients reported in Figure 10, but these plots of course don't handle the number of training tokens. In this setting, EDV's interplay

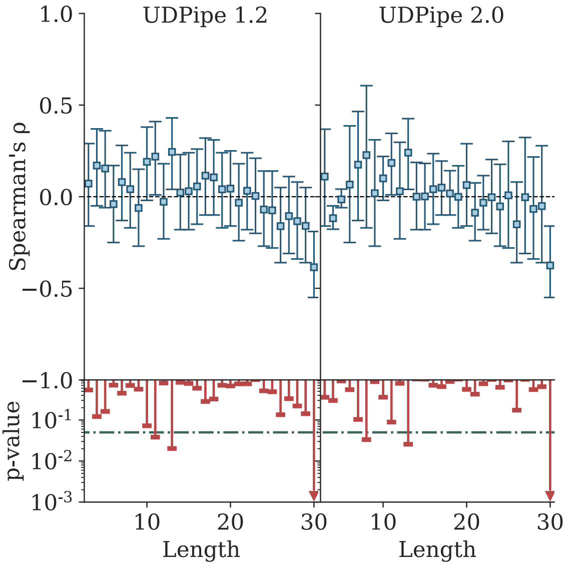

Figure 10
Partial correlation coefficients (top, blue) and their corresponding p-values (bottom, red) for UDPipe 1.2 (left) and UDPipe 2.0 for sub-samples binned with respect to sentence length. Comparison is between the EDV and LAS of each sub-sample with training tokens as covariant.

with parsing performance is not as convincing as in the other analyses. This could be related to issues with having less data as a necessary result of the binning procedure, therefore impacting the reliability of the statistics. It could also be that it introduces wider variances with respect to amount of training data in each sentence bin. It is clear that sentences of length 30 are correlated and have a coefficient that follows the trend, so it isn't as if EDV is completely uncorrelated in this setting. Whatever the reason for the different result observed here, this highlights the need to evaluate these exploratory correlation-based studies in different ways, so as to temper the certainty with which we present our results.

We note that unlike other treebank analyses focusing on measurements that are likely to be related to sentence length, EDV has a clear global correlation with our target variable (e.g., Anderson and Gómez-Rodríguez [2020] did not observe a global correlation in their analyses). But the sentence length binning highlights that different signals can be found in a more fine-grained analysis.

5. Morphological Complexity

Here we offer a small analysis of a subset of the data that are measured to be morphologically complex. We use an aggregate measurement that is explained in detail in Appendix C to measure the morphological complexity of the training data in a given treebank. This consists of 5 metrics that have been normalized and calibrated such that for each measurement 0 means no morphological complexity and 1 means maximum complexity. The average is then taken of these 5 metrics. They are based on word entropy (Shannon 1948), type–token ratio (Bentz et al. 2016), form to lemma ratio, form to inflected lemma ratio, and head part-of-speech entropy (Dehouck and Denis 2018). They all measure slightly different aspects of morphological production, except head part-of-speech entropy, which measures morphosyntactic complexity. Mathematical descriptions of these measurements are given in Appendix C, detailing the original measurements and how they have been normalized so that they could be more readily combined. For more details on these measurements (including experiments evaluating the interplay between them and parsing), see Bentz et al. (2016), Dehouck and Denis (2018), and Dehouck (2019).

We simply take the most morphologically complex treebanks by considering a treebank **morphologically complex** if its complexity is greater than the mean measurement across treebanks. This results in 50 **morphologically complex** treebanks in UD v2.5 (out of 94) and 47 in UD v2.6 (out of 90). Lists containing the specific treebanks considered morphologically complex are given in Appendix C. We cut it this way as we did not find other reasonable arguments for applying a different threshold. They are all equally arbitrary. At least following this criterion we split in a way that doesn't introduce any biases (outside of the data). It does result in some treebanks from similar languages (or the same) appearing in different subsets, for example, Portuguese-GSD has a result of 0.60 for the aggregate score, Portuguese-Bosque has 0.52, Galician-TreeGal has 0.55, and the mean score is 0.57, which results in Portuguese-GSD being classed as morphologically complex and Galician-TreeGal and Portuguese-Bosque as not. However, this measurement is not meant to classify languages but to compare given samples of a language that appear in treebanks. Furthermore, if a given property (in our particular case, correlation between LAS and EDV) tends to hold for morphologically complex treebanks and not the others (or vice versa), the fact that a treebank of intermediate complexity falls on one or other side of the split should have little influence on the

Table 6
Partial coefficients (except for rows with None in covariant column) for the full set of treebanks (Full), the morphologically complex subset (Com.), and the not morphologically complex subset (Not) for EDV with respect to LAS for UDPipe 1.2 and UD v2.5 (top) and for UDPipe 2.0 and UD v2.6 (bottom). Shown is the coefficient itself (ρ), the 95% confidence interval (CI95%), ρ^2 as an indication of the proportion of explained variance, the adjusted ρ^2 (Adj. ρ^2) as a less biased version of ρ^2, the corresponding p-values, and the achieved power of the test (power).

Parser	Set	N	**Covar.**	ρ	**CI95%**	ρ^2	**Adj.** ρ^2	**p-value**	**power**
UDPipe 1.2	Com.	50	None	−0.678	[−0.80 −0.49]	0.459	0.448	<0.001	1.000
	Not	44	None	−0.073	[−0.36 0.23]	0.005	−0.018	0.636	0.076
	Full	94	None	−0.492	[−0.63 −0.32]	0.242	0.234	<0.001	0.999
	Com.	50	Train Toks	−0.481	[−0.67 −0.23]	0.231	0.199	<0.001	0.948
	Not	44	Train Toks	0.163	[−0.14 0.44]	0.027	−0.021	0.296	0.182
	Full	94	Train Toks	−0.265	[−0.44, −0.06]	0.070	0.050	0.010	0.735
UDPipe 2.0	Com.	47	None	−0.624	[−0.77 −0.41]	0.389	0.376	<0.001	0.998
	Not	43	None	−0.118	[−0.40 0.19]	0.014	−0.010	0.450	0.118
	Full	90	None	−0.466	[−0.61 −0.29]	0.217	0.208	<0.001	0.997
	Com.	47	Train Toks	−0.466	[−0.64 −0.16]	0.183	0.146	0.003	0.857
	Not	43	Train Toks	−0.008	[−0.31 0.30]	0.000	−0.050	0.958	0.050
	Full	90	Train Toks	−0.290	[−0.47 −0.09]	0.084	0.063	0.006	0.796

aggregate metrics that we use to detect this, as long as clear-cut cases are assigned to the correct subset.

Table 6 gives the correlation coefficients for the two subsets of the data, the morphologically complex and the not morphologically complex, along with those for the full data. It is very clear that the morphologically complex subset has the clearest association with parsing performance for both parsers with an adjusted ρ^2 of 0.448 for UDPipe 1.2 and 0.376 for UDPipe 2.0, whereas the not morphologically complex subset has a negative ρ^2 for both (signaling that there is no linear relation). This clear relation holds even when accounting for the size of the training data with an adjusted ρ^2 of 0.199 for UDPipe 1.2 and 0.146 for UDPipe 2.0. It is clear from the visualization in Figure 11 that this is due to the morphologically complex subset having a wider range of EDV values with many having much higher EDV values than the small values exclusively observed from the not morphologically complex subset. It is also very clear from this visualization that it is not necessarily the case that a morphologically complex treebank will exhibit large discrepancies between samples with respect to the edge displacement distributions, i.e., many treebanks in the morphologically complex subset have very small EDV values.

6. EDV for Evaluation

Having established that EDV does correlate to parsing performance when accounting for covariants in a number of ways, we turn to a proof of concept for a potential application of EDV in NLP: using it to inform a more linguistically motivated means of creating *adversarial* splits. We note here that large EDV values between samples for a given language likely capture a linguistic feature of that language, in that large samples

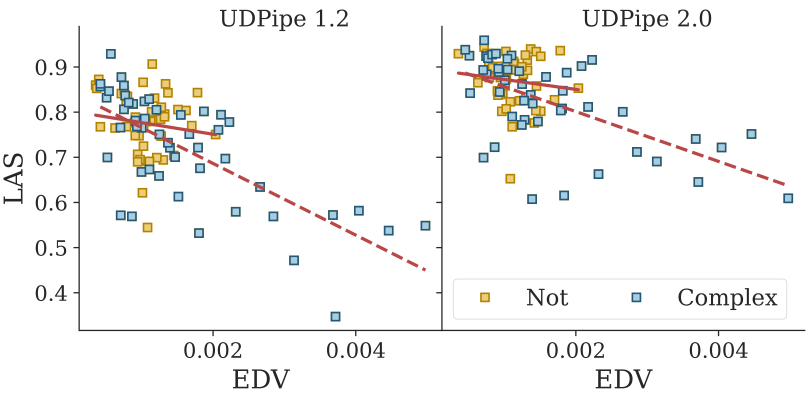

Figure 11
Visualization of LAS against EDV for UDPipe 1.2 and UDPipe 2.0 for the morphologically complex subset of treebanks (Complex, blue) and the not morphologically complex subset (Not, yellow). Linear fits are shown to aid visualization (continuous line for not complex and dashed for complex).

that deviate to a great extent suggest that language is more syntactically volatile than others. This could be true across the board or it could be a matter of greater variety in syntactic structures in a given domain. However, differences can also occur intra-domain based on author preferences.

We mention this here because recent work on developing adversarial splits focused on sentence lengths (Søgaard et al. 2021). This was an extension from criticism based on using standard splits, where random splits were suggested instead (Gorman and Bedrick 2019). Together these analyses showed that standard splits and random splits are not enough to truly evaluate the brittle nature of NLP systems trained on data from a narrow set of domains. Søgaard et al. (2021) found that even when evaluating systems with adversarial splits (based on sentence length), the evaluation overestimated the performance of the systems when compared with fresh samples. We argue that creating adversarial splits based on sentence length is only weakly linguistically motivated (i.e., the variance in sentence length could be associated with different domains, but maximizing the difference between test and training set is only a very coarse approximation of differences in domain, as not all long sentences are necessarily harder for a model to handle). With this in mind, we propose using EDV to guide sampling to create adversarial and complementary splits to give an approximation of the volatility of parsing performance. As highlighted by Søgaard et al. (2021), this only offers us a clearer picture of the generalizability of models based on the data available, which often overestimates the quality of models. However, in lieu of fresh data, this offers us a clear path to a more robust evaluation.

This approach is not dependent on the parser being evaluated as the parser does not directly play a role in developing the splits. Despite this, it is clear that splitting on EDV might not offer robust evaluation for *all* parsers and *all* parser types (e.g, parsers that are not data-driven might be less sensitive to EDV). But if that were to be the case (certain parsers being less sensitive to differences in EDV), it could be argued that such parsers *are* more robust than others. Finally, although we suggest this sampling method for evaluating parsers (and potentially other NLP systems), we are not suggesting this sampling method be the *only* means of evaluating the generalizability of models.

6.1 Sampling

We sample data in such a way so as to minimize EDV and maximize EDV, in order to give certain empirical limits of performance for a given treebank. We do this by collating all trees for a given treebank across all splits that are available. We remove trees with 2 tokens or less. We then bin the trees by sentence length and by the mean edge displacement (MED) of each sentence. MED is defined as:

$$\text{MED} = \frac{1}{N-1} \sum_{n \in N} s^{n}_{edge} \tag{3}$$

where n is a given node in a given tree, N is the total number of nodes in the tree, and s^{n}_{edge} is the edge displacement of a given node as defined in Equation (1). Note the denominator is $N - 1$ as the root node is not included.

We initialize the process by selecting a sentence length at random and also an MED value that exists for that sentence length bin. We then sample 3 more sentences with the closest sentence length and closest MED value available. This gives us 4 sentences with the same (or similar) sentence length and the same (or similar) MED. These are added to the training trees. We then either sample a sentence to match the MED value (when trying to keep EDV low) or sample a sentence with the furthest MED value available for the current sentence length bin (or closest if no sentences are left in a given bin) in order to maximize EDV. We repeat this process with the subsequent MED values chosen for the training instances to match the overall MED of the current training data. We do this until we have split the whole data into 80% training data and 20% test data. We then split the training data so as to obtain development data such that the overall split is 60|20|20 for training, dev, and test data, respectively. Note, we use MED and sample by tree as a more direct use of EDV would require the creation of many samples and hoping that one serendipitously maximizes/minimizes EDV. One could also potentially use an evolutionary algorithm to find splits that maximize (or minimize) EDV, but it would likely be computationally expensive.

We then train models using UDPipe 1.2 for the minimized EDV split and the maximized EDV split. We do this for all treebanks that have a training set of 100 sentences or more in the original split.

6.2 Sampling Results

Figure 12 shows the distributions of ΔLAS (LAS$_{max}$$-LAS_{min}$) for each treebank. We fit the distribution with a skewed Gaussian function to better evaluate variance seen in this process (a more conservative one at least). When evaluating the mean of the data itself we see a mean ΔLAS of -4.26 (2.17), whereas the fit is slightly lower and with a higher standard deviation at -4.18 (2.68). This difference is considerable with typical claims of state-of-the-art performance coming down to tenths of a LAS point, so this process certainly gives a good range of performance across treebanks. Figure 13 shows the actual distribution of LAS values for both sets of splits. The median values of LAS are 75.40 and 70.77 for the minimum and maximum EDV splits, respectively. Note too that the spread across the first and second quartile is wider for the maximum EDV splits and with the lower tail being much smaller than that of the minimum split.

We also evaluate whether this difference in performance can be attributed to the differences in EDV between the splits. Figure 14 shows ΔLAS against ΔEDV

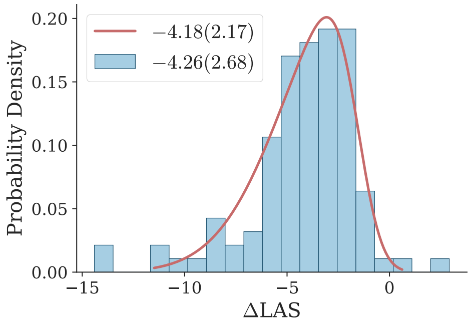

Figure 12
Distribution of ΔLAS (the LAS obtained from split where EDV is minimized minus the LAS obtained for the split where EDV is maximized) for UDPipe 1.2 models trained using UD v2.7 (103 treebanks). Shown is a fit used to obtain a more conservative measure of the variance between splits with $\chi^2 = 0.40$ and p-value $= 0.820$ (note H_0 means the data comes from the distribution described by the fit).

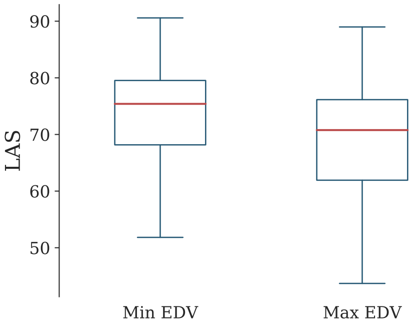

Figure 13
Distribution of LAS for UDPipe 1.2 models trained on splits sampled so as to miminize EDV (Min EDV) and sampling so as to maximize EDV (Max EDV) using UD v2.7 (103 treebanks). The median LAS for Min EDV is 75.40 and 70.77 for Max EDV.

($\mathrm{EDV}_{max} - \mathrm{EDV}_{min}$). A strong negative linear relationship is observed, as expected. To validate this observation, we once again turn to correlation coefficients. These are reported in Table 7. We look at the variables deemed most pertinent to evaluate EDV from the preceding analysis in Section 4.2. In this context, the number of training tokens (here we take the mean across the splits as an approximation) is not associated with the difference in performance across splits, which would only likely be the case if this had a major role in constraining the maximization of EDV. Similarly, the difference between the number of training tokens is not correlated to ΔLAS (the difference between splits is not large at a mean relative difference of 0.097%).

However, $\langle L_{\text{test}} \rangle$ (defined as the mean across splits) is strongly correlated to ΔLAS (ρ = 0.507, p-value $<$ 0.001) and even more so to ΔEDV (ρ = 0.847, p-value $<$ 0.001). This is likely due to the dependence on sentence length to vary EDV (see Figure 3 and Figure A.1 in Appendix A). However, the difference between $\langle L_{\text{test}} \rangle$ for each split is not correlated to ΔLAS, meaning that the difference observed is not merely due to the sampling procedure being forced to sample sentences of different length so as to

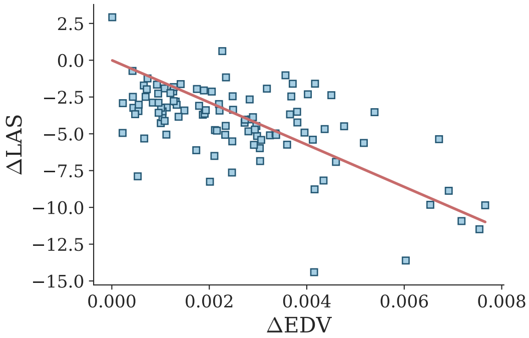

Figure 14
ΔLAS against ΔEDV where both are the value associated with the split where EDV has been minimized minus that of the split where EDV has been maximized, for UDPipe 1.2 models using UD v2.7 (103 treebanks).

Table 7
Correlations between variables of interest with respect to ΔLAS using UD v2.7 (103 treebanks) and UDPipe 1.2. Shown are the coefficients (ρ), the 95% confidence intervals (CI95%), ρ^2 as an indication of the proportion of explained variance, the adjusted ρ^2 (Adj. ρ^2) as a less biased version of ρ^2, the corresponding p-values, and the achieved power of the tests (power). The mean absolute ΔTokens is 45.0 (68.7), which is a relative difference 0.097% (with respect to the split where EDV is minimized). Mean absolute $\Delta\langle L_{\text{test}}\rangle$ is 0.059 (0.172), which is a relative difference of 0.25% with respect to. Tokens and $\langle L_{\text{test}}\rangle$ used here are the average across both splits.

Variable	**Target**	**Covar.**	ρ	**CI95%**	ρ^2	**Adj.** ρ^2	**p-value**	**power**
Train Tokens	ΔLAS	—	0.104	[−0.09, 0.29]	0.011	0.001	0.295	0.182
$\langle L_{\text{test}}\rangle$			0.507	[0.35, 0.64]	0.258	0.251	<0.001	1.000
ΔTokens			0.067	[−0.13, 0.26]	0.004	−0.006	0.502	0.103
$\Delta\langle L_{\text{test}}\rangle$			−0.037	[−0.23, 0.16]	0.001	−0.009	0.713	0.065
ΔSLV			0.139	[−0.06 0.32]	0.019	0.010	0.161	0.290
ΔEDV			−0.478	[−0.61, −0.31]	0.228	0.220	<0.001	0.999
$\langle L_{\text{test}}\rangle$	ΔEDV	—	−0.847	[−0.89, −0.78]	0.717	0.711	<0.001	1.000
ΔSLV	ΔEDV		0.088	[−0.11 0.28]	0.008	−0.002	0.379	0.143
ΔEDV	ΔLAS	$\langle L_{\text{test}}\rangle$	−0.105	[−0.29 0.09]	0.011	−0.009	0.295	0.182
ΔEDV	ΔLAS	ΔSLV	−0.497	[−0.63 −0.33]	0.247	0.231	<0.001	1.000

maximize EDV. This is further attested to by the small mean relative difference between the splits of 0.25%.

ΔEDV is also strongly correlated to ΔLAS at −0.478 (p-value < 0.001), which fits with the trend observed in Figure 14. We also report the partial coefficient of ΔEDV with respect to ΔLAS with $\langle L_{\text{test}}\rangle$ as a covariant. This results in a coefficient of −0.105 (p-value = 0.295), which clearly shows the variation in EDV between the splits is strongly bounded by the sentence lengths of the data. In a sense, the sentence length distribution or $\langle L_{\text{test}}\rangle$ dictates how much we can optimize the difference between the max and min EDV splits, although sampling splits so as to maintain a similar sentence length distribution across splits but sampling randomly for each sentence length bin is likely to result in easier splits than maximizing EDV. We also check to see if the difference between the sentence length distributions (ΔSLV) diminishes the correlation between

ΔEDV and ΔLAS if used as a covariant, but it doesn't (it is in fact slightly larger but this increase is meaningless).

7. Conclusion

We have offered an analysis that has shown a clear correlation between the differences in the edge displacement distributions of training and test data in UD treebanks (as measured by the Vaserstein distance) and parsing performance (as measured by the labeled attachment score) by using a number of methods to falsify this hypothesis. We attempted to remove signals associated with covariants which were also correlated with LAS, but still observed a linear relationship between EDV and a normalized LAS. We use statistical methods to first evaluate the partial correlations of EDV and LAS when accounting for covariants and still observed meaningful coefficients. We also used multilinear regression to evaluate whether EDV adds any predictive power to models using these same covariants and measured small but meaningful contributions from EDV. In addition, we evaluated this linear model by training new parsers with one of the systems under investigation here on treebanks in the most recent release of UD that did not already have a model and obtained predictions that were not outlandish, especially for higher performing treebanks. Further, we evaluated the partial coefficients for EDV when using a sentence-length binning analysis and observed stronger coefficients for sentences of moderate length with a clear monotonic relationship between the magnitude of the correlation of EDV to LAS and sentence length. However, the p-values are fairly high with only the largest sentences (of 30 tokens) exhibiting a large and clear correlation to parsing performance.

As mentioned above, we suspect EDV is indicative of parsing performance because it captures syntactic differences at the sample level, which could be due to a number of reasons, spanning different syntactic structures being adopted in different domains to linguistic features of a language causing greater degrees of freedom in the tree structures found in different samples. Beyond linguistic considerations, the difference in performance observed due to EDV is likely to be explained by supervised techniques struggling to predict unobserved patterns, as larger EDV values indicate differences in the tree patterns found in the training and test data.

Finally, we have shown the potential for using EDV to create splits to evaluate an advantageous and a disadvantageous (based on the available data) scenario that is likely to be more indicative of real-world usage of parsers where out-of-domain, unseen syntactic structures likely occur in the outer regions of the distributions seen in narrow training data sets. We envisage this analysis also being useful for other practices in NLP. For example, it could be used for evaluating the difficulty of a given instance for curriculum learning for training parsers or for other NLP tasks, that is, batches measured for EDV based on the overall distribution in the training data.

Appendix A. Further Visualization

This appendix is mainly for showing the corresponding data for UDPipe 1.2 (as we showed the data for UDPipe 2.0 for the most part in the main text). Almost universally the observed behavior follows that shown in the main text. If it had been otherwise, we would have opted to show conflicting data visualizations. Figures A.7 and A.8 show the data used to evaluate the coefficients shown in Figure 10.

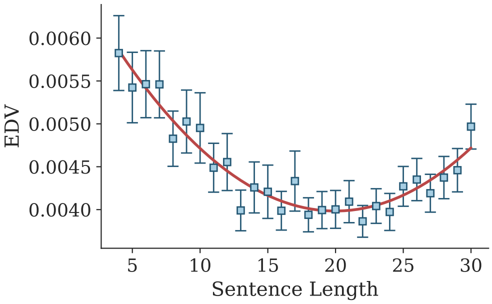

Figure A.1
EDV between sub-samples of the training and test data binned by sentence length for UD v2.5 (105 treebanks).

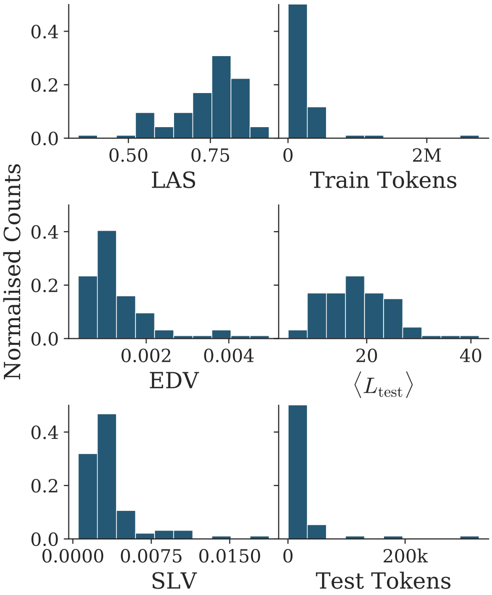

Figure A.2
Distributions of the variables of interest in UD v2.5 (94 treebanks) in order to evaluate whether they are sampled from normal distributions.

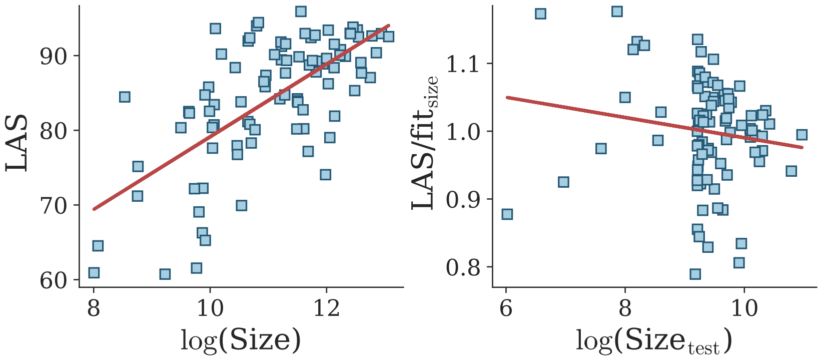

Figure A.3
Background removing method used to evaluate whether the number of test tokens carries additional information with respect to the number of training tokens for UDPipe 2.0 and UD v2.6. Correlation between the number of test tokens and LAS is 0.309 (p-value = 0.003) and that between the number of test tokens and the normalized LAS (right plot) is −0.101 (p-value = 0.342).

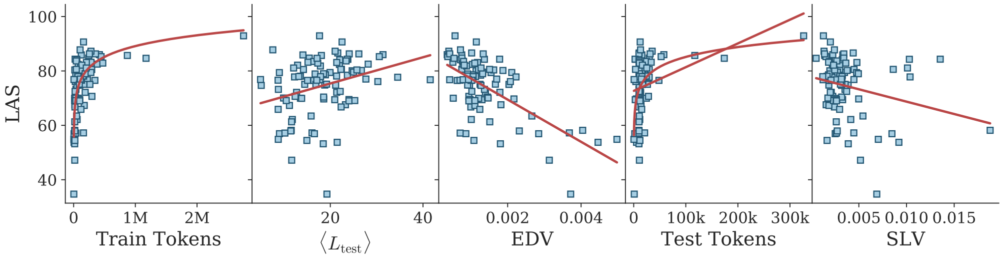

Figure A.4
Visualization of LAS (for UDPipe 1.2 and UD v2.5) with respect to variables of interest with fits shown in red to highlight whether the data appears correlated or not.

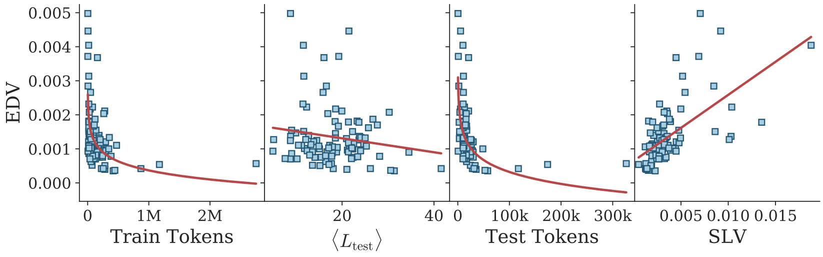

Figure A.5
Visualization of EDV (for UD v2.5) with respect to variables of interest with fits shown in red to highlight whether the data appears correlated or not.

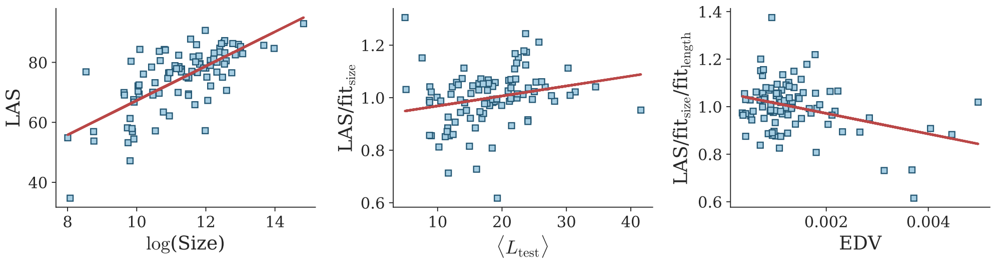

Figure A.6
Background removal method to evaluate whether a correlation is observed between EDV and LAS (for UDPipe 1.2 and UD v2.5) after removing the variation associated with the training test size and $\langle L_{\text{test}} \rangle$. The correlation between EDV and LAS is -0.492 (p-value < 0.001), the correlation between EDV and the LAS normalized by the variance associated with number of tokens in training data is -0.186 (p-value $= 0.072$), and the correlation for the fully normalized LAS (removing the variance associated with $\langle L_{\text{test}} \rangle$) is -0.249 (p-value $= 0.015$).

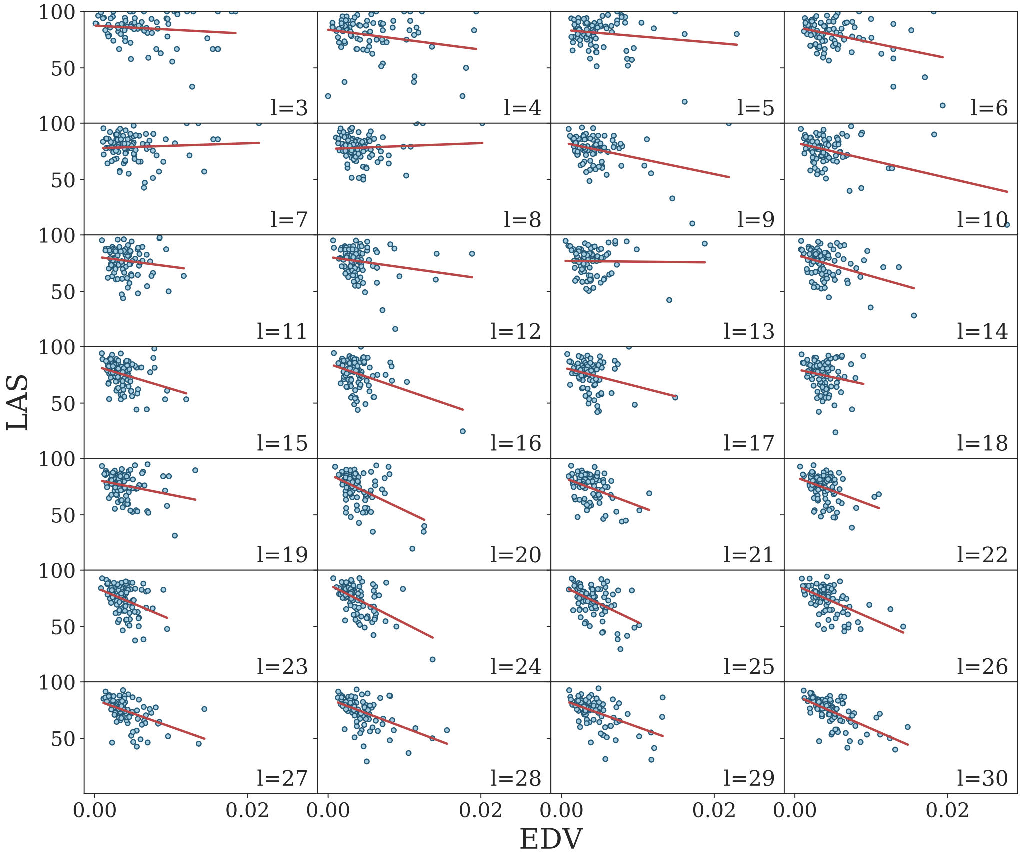

Figure A.7
LAS versus EDV for each sentence length bin (labeled l = length) for UDPipe 1.2 used for calculating the coefficients shown in Figure 10.

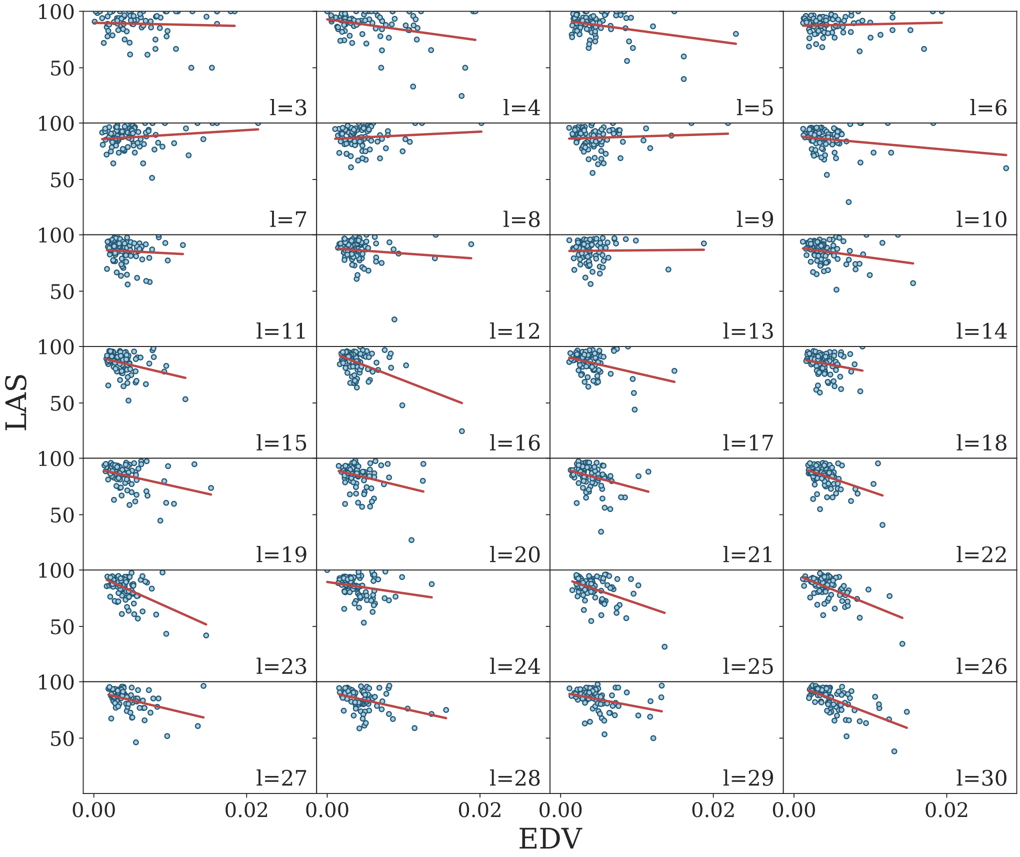

Figure A.8
LAS versus EDV for each sentence length bin (labeled l = length) for UDPipe 2.0 used for calculating the coefficients shown in Figure 10.

Appendix B. Training Data Measurements Unrelated to EDV

In this appendix, we include additional analysis looking at some linguistically focused measurements of the training data that are related to parsing performance. These are presented here rather than the main text because there is no theoretical justification for expecting these measurements to impact the EDV of a given treebank split. The first metric is a normalized count of the number of crossings in a tree $C/|Q|$, where C is the number of crossings in a tree and $|Q|$ is the total number of possible crossings (Ferrer-i Cancho, Gómez-Rodríguez, and Esteban 2018). The second measurement is the type–token ratio, defined as the number of unique forms divided by the number of tokens found in a treebank, which gives a measure of the lexical diversity of a sample and a coarse indication of the degree of morphology. As can be seen in Table B.1, neither of these measurements are correlated to EDV for either UD v2.5 or UD v2.6. A visualization of the corresponding data is shown in Figure B.1.

Table B.1
Spearman's ρ for correlations between treebank measurements of training data and EDV. $C/|Q|$ is the normalized number of crossings found in a treebank and TTR is the type–token ratio.

Data	Variable	ρ	CI95%	p-value	power
UD v2.5	$C/\|Q\|$	0.126	[−0.08, 0.32]	0.228	0.227
	TTR	−0.048	[−0.25, 0.16]	0.644	0.075
UD v2.6	$C/\|Q\|$	0.136	[−0.07, 0.33]	0.202	0.248
	TTR	−0.038	[−0.24, 0.17]	0.724	0.064

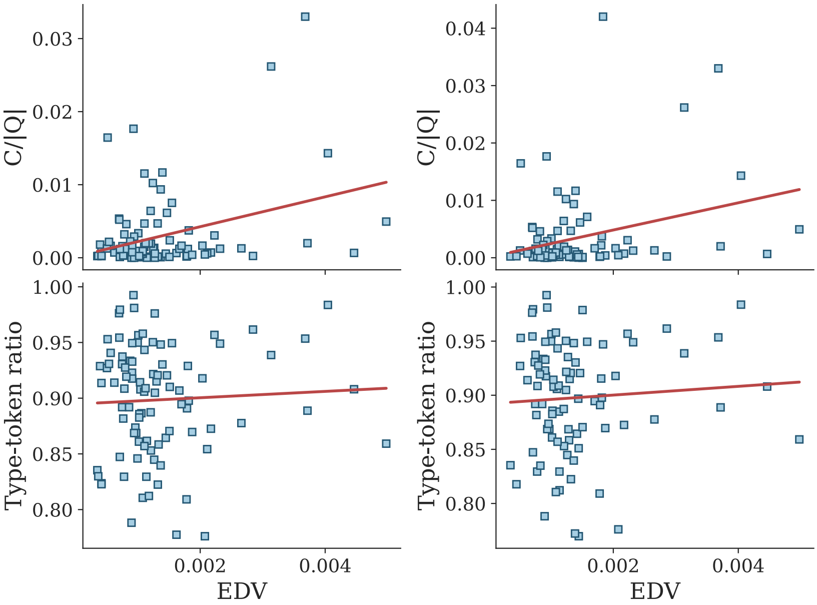

Figure B.1
$C/|Q|$ (the normalized number of crossings found in a treebank) and type–token ratio of the training data found in UD v2.5 (column 1) and UD v2.6 (column 2) against the corresponding EDV for each treebank.

Appendix C. An Aggregate Measurement of Morphological Complexity

In this appendix, we provide the details about the measurement we used for approximating morphologically complex subsets of the treebanks that were used in Section 5. It is an aggregate measurement, consisting of word entropy (Shannon 1948), type–token ratio (Bentz et al. 2016), form to lemma ratio, form to inflected lemma ratio, and head part-of-speech entropy (Dehouck and Denis 2018). These are normalized when needed such that 0 means no morphological complexity and 1 means the highest possible morphological complexity, so that we can simply take the mean measurement across all 5 metrics.

Normalized Word Entropy: Word entropy gives an indication as to how much information any given word has, with a higher entropy resulting from a treebank having many forms. It is given by:

$$H_{\text{word}} = -\sum_{v \in \mathcal{V}} p(v) \log_2 p(v) \tag{C.1}$$

where $\mathcal{V}$ is the vocab space in a given treebank, v is a given word in that space, and $p(v)$ is the probability of that word occurring estimated by its frequency count (Shannon 1948). The normalized word entropy, H^*_{word}, is obtained by dividing by the log of the magnitude of the vocab space:

$$H^*_{\text{word}} = \frac{H_{\text{word}}}{log_2|\mathcal{V}|} \tag{C.2}$$

Type–Token Ratio: The type–token ratio gives an indication of the morphological production in a given treebank. It is given by:

$$TTR = \frac{|\mathcal{V}|}{|T|} \tag{C.3}$$

where $\mathcal{V}$ is the vocab space in a given treebank and T is the number of tokens (Bentz et al. 2016). While this number isn't exactly bounded by 0 at the lower margin (it is bounded by 1 at the upper margin), when T is suitably big, which is typically the case, the instance where $\mathcal{V}$ only consists of 1 type, TTR tends to zero. However, this is clearly not a likely scenario in a treebank and so this inconsistency is not a worry in reality.

Form to Lemma Ratio: The form to lemma ratio is similar to the type–token ratio but it more closely measures morphological production by homing in on lemmas having multiple forms rather than just looking at the more global measurement of production in TTR. It is given by:

$$F/L = \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} |\mathcal{F}_l| \tag{C.4}$$

where $\mathcal{L}$ is the lemma vocab of a treebank, l is a given lemma in the vocab, and $\mathcal{F}_l$ is the set of forms associated with l (Dehouck and Denis 2018).

As defined, F/L ranges from 1 to $|\mathcal{V}|$ (the absurd case of a singular lemma). By taking the reciprocal, we obtain a value that tends to zero in the absurd case and has an upper bound of 1. However, this gives us an inverse scale, that is, a lower value means more morphology and a higher value less. Therefore we subtract the reciprocal of F/L from 1:

$$F/L^* = 1 - \frac{1}{F/L} \tag{C.5}$$

Inflected Form to Lemma Ratio: This is the same as F/L but for the case where a lemma is actually inflected, namely, the case where the set of word forms associated with a given lemma is greater than 1. It is given by:

$$F/iL = \frac{1}{|\mathcal{L}_2|} \sum_{l \in \mathcal{L}_2} |\mathcal{F}_l| \tag{C.6}$$

where $\mathcal{L}_2$ is the subset of lemmas that have 2 or more forms associated with them in a treebank, l is a given lemma in that subset, and $\mathcal{F}_l$ is the set of forms associated with l (Dehouck and Denis 2018). It is normalized in the same way as F/L:

$$Fi/L^* = 1 - \frac{1}{F/iL} \tag{C.7}$$

Head Part-of-Speech Entropy: The head part-of-speech entropy (HPE) is the measurement of morphology most related to parsing as it captures the morphosyntactic complexity found in a treebank. It is measured on the delexicalized version of the treebank, where the unit is a concatenation of a token's POS tag and morphological feature tags. The HPE of a treebank is an average over the HPE of each delexicalized word type:

$$HPE = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} HPE_d \tag{C.8}$$

where:

$$HPE_d = -\sum_{t \in \mathcal{T}_d} p(h_d^t) \log_2 p(t_d^t) \tag{C.9}$$

where h_d^t denotes the head of d having the POS tag t from the tagset $\mathcal{T}_d$ (the set of tags that d is headed by in the treebank) and $p(h_d^t)$ is the probability of this occurring based on frequency counts (Dehouck and Denis 2018). As defined this gives a value that tends to zero when morphosyntactic complexity is prevalent and increases unbounded the less morphosyntactic complexity is present. In order to normalize this, we have to normalize HPE_d:

$$HPE_d^* = \frac{HPE_d}{\log_2 |\mathcal{T}_d|} \tag{C.10}$$

such that the normalized head part-of-speech entropy is simply:

$$HPE^* = 1 - \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} HPE_d^* \tag{C.11}$$

Note that the sum over the normalized HPE_d values is subtracted from 1 to invert the scale such that 0 denotes no morphosyntactic complexity and 1 the maximum.

Aggregate metric: The final metric we used is a simple unweighted average of the 5 normalized metrics described above:

$$MC = \frac{(H^*_{\text{word}} + TTR + F/L^* + F/iL^* + HPE^*)}{5} \tag{C.12}$$

Below are the lists with the treebanks considered *morphologically complex* in UD v2.5 and v2.6, respectively. The code used for this is available at `https://github.com/markda/morphological-complexity`.

List of morphologically complex treebanks in UD v2.5:

Ancient Greek-PROIEL
Ancient Greek-Perseus
Armenian-ArmTDP
Basque-BDT
Belarusian-HSE
Bulgarian-BTB
Croatian-SET
Czech-CAC
Czech-CLTT
Czech-FicTree
Czech-PDT
Estonian-EDT
Estonian-EWT
Finnish-FTB
Finnish-TDT
German-HDT
Gothic-PROIEL
Greek-GDT
Hungarian-Szeged
Irish-IDT
Latin-ITTB
Latin-PROIEL
Latin-Perseus
Latvian-LVTB
Lithuanian-ALKSNIS
Lithuanian-HSE
Maltese-MUDT
Marathi-UFAL
North Sami-Giella
Old Church Slavonic-PROIEL
Old French-SRCMF
Old Russian-TOROT
Persian-Seraji
Polish-LFG
Polish-PDB
Portuguese-GSD
Romanian-Nonstandard
Romanian-RRT
Russian-GSD
Russian-SynTagRus
Russian-Taiga
Serbian-SET
Slovak-SNK
Slovenian-SSJ
Slovenian-SST
Tamil-TTB
Telugu-MTG
Turkish-IMST

List of morphologically complex treebanks in UD v2.6:

Ancient Greek-PROIEL
Ancient Greek-Perseus
Armenian-ArmTDP
Basque-BDT
Belarusian-HSE
Bulgarian-BTB
Croatian-SET
Czech-CAC
Czech-CLTT
Czech-FicTree
Estonian-EDT
Estonian-EWT
Finnish-FTB
Finnish-TDT
Gothic-PROIEL
Greek-GDT
Hungarian-Szeged
Irish-IDT
Latin-ITTB
Latin-PROIEL
Latin-Perseus
Latvian-LVTB
Lithuanian-ALKSNIS
Lithuanian-HSE
Maltese-MUDT
Marathi-UFAL
North Sami-Giella
Old Church Slavonic-PROIEL
Old French-SRCMF
Old Russian-RNC
Old Russian-TOROT
Persian-Seraji
Polish-LFG
Polish-PDB
Portuguese-GSD
Romanian-RRT
Russian-GSD
Russian-Taiga
Sanskrit-Vedic
Serbian-SET
Slovak-SNK
Slovenian-SSJ
Slovenian-SST
Tamil-TTB
Telugu-MTG

Appendix D. Variance in EDV for Different Sizes of Training Data

We offer a small analysis on how much variance we observe in the same treebank when sampling smaller amounts of training data, as it is possible that evaluation undertaken in this work would only hold true for these very specific splits—although the sampling evaluation makes this unlikely.

We selected treebanks from UD v2.7 that had at least 20,000 training instances, so that samples could be sufficiently different. We opted for Czech-PDT, Estonian-EDT,

Table D.1
Standard deviation is reported with respective means in the form mean (standard deviation). Each value x (other than full training count) in the table corresponds to $x \times 10^{-4}$. Values are given for sample sizes of 2K, 4K, 6K, and 8K training sentences. The final column "Full training" gives the total number of sentences in the original training data. Standard deviation ranges from 3.7% to 17.1% of respective means (5.6% on average).

	EDV — mean$\times 10^{-4}$ (standard deviation$\times 10^{-4}$)				
	2K Sample	4K Sample	6K Sample	8K Sample	Full training
Czech-PDT	3.8 (0.7)	3.4 (0.5)	3.2 (0.3)	3.3 (0.4)	68,495
Estonian-EDT	5.8 (0.7)	5.9 (0.6)	5.6 (0.4)	5.4 (0.3)	24,633
German-HDT	4.1 (0.7)	3.6 (0.5)	3.3 (0.4)	3.3 (0.3)	153,035
Japanese-BCCWJ	9.9 (0.9)	10.0 (0.7)	9.6 (0.5)	9.8 (0.5)	40,740
Korean-Kaist	5.4 (0.5)	5.3 (0.5)	5.2 (0.3)	5.0 (0.2)	23,010
Persian-PerDT	6.2 (0.8)	5.8 (0.5)	5.8 (0.5)	5.9 (0.3)	26,196

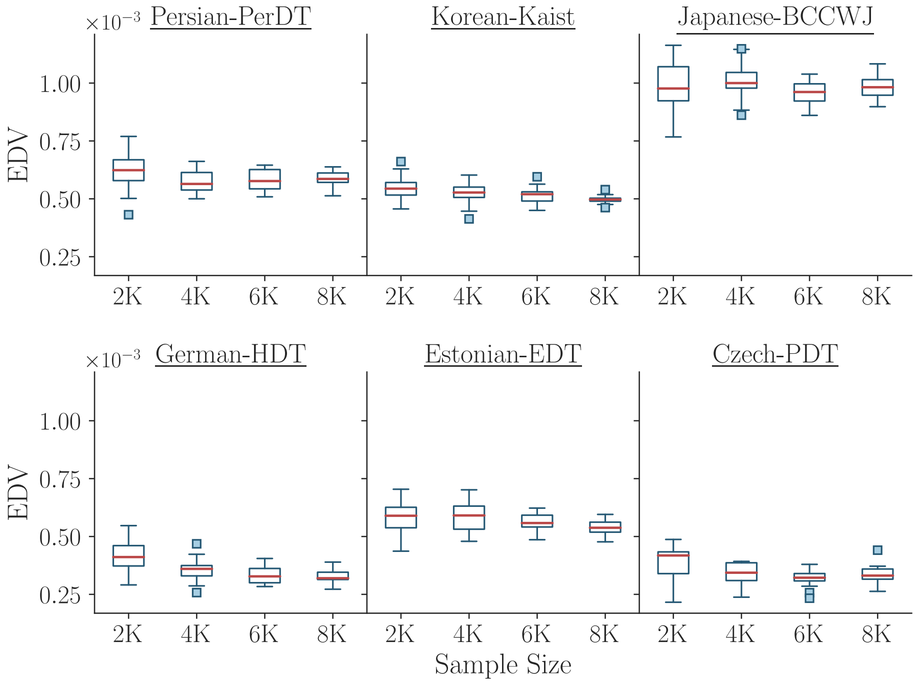

Figure D.1
Distributions of EDV for different treebanks with varying sample sizes. Smaller sample sizes exhibit greater variance than larger sample sizes, but not to such a degree that the values measured for EDV for different languages change how they compare to those of other treebanks.

German-HDT, Japanese-BCCWJ, Korean-Kaist, and Persian-PerDT, as this offered us the best spread across different languages and language families from the largest treebanks. We then sampled different amounts of training data for each of these treebanks. We sampled 2,000; 4,000; 6,000; and 8,000 training samples. For each training size, we sampled 20 unique sets of training instances.

Figure D.1 shows the distributions of EDV values split across each treebank for the different training size. The first thing that is clear is that across different sample sizes, the differences observed across treebanks are fairly stable. However, for most treebanks the variance in EDV decreases as the sample size increases. This is expected and the variance observed for the smallest sample size is still not particularly high. Table D.1 gives the corresponding mean and standard deviation from the values used in Figure D.1. This further corroborates that the variance in EDV for the smaller sample sizes is not problematically high, with the standard deviation averaging at 5.6% of their respective means.

Acknowledgments

This work has received funding from the European Research Council (ERC), under the European Union's Horizon 2020 research and innovation program (FASTPARSE, grant agreement no. 714150), from ERDF/MICINN-AEI (ANSWER-ASAP, TIN2017-85160-C2-1-R; SCANNER-UDC, PID2020-113230RB-C21), from Xunta de Galicia (ED431C 2020/11), and from Centro de Investigación de Galicia "CITIC", funded by Xunta de Galicia and the European Union (ERDF - Galicia 2014-2020 Program), by grant ED431G 2019/01.

References

Alicante, Anita, Cristina Bosco, Anna Corazza, and Alberto Lavelli. 2012. A treebank-based study on the influence of Italian word order on parsing performance. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 1985–1992.

American Psychological Association. 2010. *Publication Manual of the American Psychological Association*, 6th edition. American Psychological Association, Washington, DC.

Anderson, Mark 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.

Anderson, Mark, Anders Søgaard, and Carlos Gómez-Rodríguez. 2021. Replicating and extending "Because their treebanks leak": Graph isomorphism, covariants, and parser performance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1090–1098. https://doi.org/10.18653/v1/2021.acl-short.138

Bentz, Christian, Tatjana Soldatova, Alexander Koplenig, and Tanja Samardžić. 2016. A comparison between morphological complexity measures: Typological data vs. language corpora. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*, pages 142–153.

Berdicevskis, Aleksandrs, Çagri 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. *Second Workshop on Universal Dependencies*, pages 8–17. https://doi.org/10.18653/v1/W18-6002

Bosco, Cristina, Simonetta Montemagni, Alessandro Mazzei, Vincenzo Lombardo, Felice Dell'Orletta, Alessandro Lenci, Leonardo Lesmo, Giuseppe Attardi, Maria Simi, Alberto Lavelli, Johan Hall, Jens Nilsson, and Joakim Nivre. 2010. Comparing the influence of different treebank annotations on dependency parsing. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 1794–1801.

Buch-Kromann, Matthias. 2006. *Discontinuous Grammar: A Dependency-Based Model of Human Parsing and Language Learning*. Ph.D. thesis, Copenhagen Business School.

Chen, Danqi and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750.

Chu, Yoeng Jin and Tseng-Hong Liu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400.

Chung, Tagyoung, Matt Post, and Daniel Gildea. 2010. Factors affecting the accuracy of Korean parsing. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 49–57.

Cöltekin, Çağrı. 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.

Corazza, Anna, Alberto Lavelli, and Giorgio Satta. 2013. An information-theoretic measure to evaluate parsing difficulty across treebanks. *ACM Transactions on Audio, Speech, and Language Processing*, 9(4). `https://doi.org/10.1145/2407736.2407737`

Dehouck, Mathieu. 2019. *Multi-Lingual Dependency Parsing: Word Representation and Joint Training for Syntactic Analysis. (Parsing en Dépendances Multilingue: Représentation de Mots et Apprentissage Joint pour l'Analyse Syntaxique).* Ph.D. thesis, Université de Lille. `https://doi.org/10.18653/v1/N19-1017`

Dehouck, Mathieu, 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. `https://doi.org/10.18653/v1/2020.iwpt-1.20`

Dehouck, Mathieu 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. `https://doi.org/10.18653/v1/D18-1312`

Dehouck, Mathieu and Carlos Gómez-Rodríguez. 2020. Data augmentation via subtree swapping for dependency parsing of low-resource languages. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3818–3830. `https://doi.org/10.18653/v1/2020.coling-main.339`

Dozat, Timothy and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR* 2017, Conference Track Proceedings, OpenReview.net.

Edmonds, Jack. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards B*, 71(4):233–240. `https://doi.org/10.6028/jres.071B.032`

Eisner, Jason and Noah A. Smith. 2010. Favor short dependencies: Parsing with soft and hard constraints on dependency length. In *Trends in Parsing Technology*. Springer, pages 121–150. `https://doi.org/10.1007/978-90-481-9352-3_8`

Falenska, Agnieszka, 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. `https://doi.org/10.18653/v1/2020.iwpt-1.4`

Falenska, Agnieszka 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.

Farrar, Donald E. and Robert R. Glauber. 1967. Multicollinearity in regression analysis: The problem revisited. *Review of Economic and Statistics*, 49(1):92–107. `https://doi.org/10.2307/1937887`

Ferrer-i-Cancho, Ramon. 2004. Euclidean distance between syntactically linked words. *Physical Review E*, 70(5):056135. `https://doi.org/10.1103/PhysRevE.70.056135`, PubMed: 15600720

Ferrer-i-Cancho, Ramon and Carlos Gómez-Rodríguez. 2016. Crossings as a side effect of dependency lengths. *Complexity*, 21(S2):320–328. `https://doi.org/10.1002/cplx.21810`

Ferrer-i Cancho, Ramon, Carlos Gómez-Rodríguez, and Juan Luis Esteban. 2018. Are crossing dependencies really scarce? *Physica A: Statistical Mechanics and its Applications*, 493:311–329. `https://doi.org/10.1016/j.physa.2017.10.048`

Ferrer-i-Cancho, Ramon and Haitao Liu. 2014. The risks of mixing dependency lengths from sequences of different length. *Glottotheory*, 5(2):143–155. `https://doi.org/10.1515/glot-2014-0014`

Foster, Jennifer. 2010. "cba to check the spelling": Investigating parser performance on discussion forum posts. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 381–384.

Futrell, Richard, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 languages. In *Proceedings of the National Academy of Sciences*, 112(33):10336–10341. https://doi.org/10.1073/pnas.1502134112, PubMed: 26240370

Gibson, Edward. 2000. The dependency locality theory: A distance-based theory of linguistic complexity. *Image, Language, Brain*, 2000:95–126.

Gildea, Daniel. 2001. Corpus variation and parser performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 167–202.

Gildea, Daniel and David Temperley. 2010. Do grammars minimize dependency length? *Cognitive Science*, 34(2):286–310. https://doi.org/10.1111/j.1551-6709.2009.01073.x, PubMed: 21564213

Gómez-Rodríguez, Carlos. 2017. On the relation between dependency distance, crossing dependencies, and parsing. *Physics of Life Reviews*, 21:200–203. https://doi.org/10.1016/j.plrev.2017.05.007, PubMed: 28595849

Gómez-Rodríguez, Carlos and Ramon Ferrer-i-Cancho. 2017. Scarcity of crossing dependencies: A direct outcome of a specific constraint? *Physical Review E*, 96(6):062304. https://doi.org/10.1103/PhysRevE.96.062304, PubMed: 29347395

Gorman, Kyle and Steven Bedrick. 2019. We need to talk about standard splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2786–2791. https://doi.org/10.18653/v1/P19-1267

Groemping, Ulrike. 2006. Relative importance for linear regression in R: The package relaimpo. *Journal of Statistical Software*, 17(1):1–27. https://doi.org/10.18637/jss.v017.i01

Gulordava, Kristina and Paola Merlo. 2015. Diachronic trends in word order freedom and dependency length in dependency-annotated corpora of Latin and Ancient Greek. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 121–130.

Gulordava, Kristina and Paola Merlo. 2016. Multi-lingual dependency parsing evaluation: A large-scale analysis of word order properties using artificial data. *Transactions of the Association for Computational Linguistics*, 4:343–356. https://doi.org/10.1162/tacl_a_00103

Hudson, Richard. 2017. Cross-language diversity, head-direction and grammars. Comment on "Dependency distance: A new perspective on syntactic patterns in natural languages" by Haitao Liu et al. *Physics of Life Reviews*, 21:204–206. https://doi.org/10.1016/j.plrev.2017.06.005, PubMed: 28602718

Jiang, Jingyang and Haitao Liu. 2015. The effects of sentence length on dependency distance, dependency direction and the implications–Based on a parallel English–Chinese dependency treebank. *Language Sciences*, 50:93–104. https://doi.org/10.1016/j.langsci.2015.04.002

Kulmizev, Artur, Miryam de Lhoneux, Johannes Gontrum, Elena Fano, and Joakim Nivre. 2019. Deep contextualized word embeddings in transition-based and graph-based dependency parsing—A tale of two parsers revisited. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2755–2768. https://doi.org/10.18653/v1/D19-1277

Kübler, Sandra, Ines Rehbein, and Josef van Genabith. 2008. A testsuite for testing parser performance on complex German grammatical constructions. *LOT Occasional Series*, 12:15–28.

de Lhoneux, Miryam, Sara Stymne, and Joakim Nivre. 2017. Old school vs. new school: Comparing transition-based parsers with and without neural network enhancement. In *Proceedings of the 15th Treebanks and Linguistic Theories Workshop (TLT)*, pages 99–110.

Liu, Haitao. 2007. Probability distribution of dependency distance. *Glottometrics*, 15:1–12.

Liu, Haitao. 2008. Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*, 9(2):159–191. https://doi.org/10.17791/jcs.2008.9.2.159

Liu, Haitao, Chunshan Xu, and Junying Liang. 2017. Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of Life Reviews*, 21:171–193. https://doi.org/10.1016/j.plrev.2017.03.002, PubMed: 28624589

Matsuzaki, Takuya and Jun'ichi Tsujii. 2008. Comparative parser performance analysis across grammar frameworks through automatic tree conversion using synchronous grammars. In *Proceedings of the 22nd International Conference on*

Computational Linguistics (Coling 2008), pages 545–552. https://doi.org/10.3115/1599081.1599150

McCarthy, Arya D., Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244. https://doi.org/10.18653/v1/W19-4226

McDonald, Ryan and Joakim Nivre. 2011. Analyzing and integrating dependency parsers. *Computational Linguistics*, 37(1):197–230. https://doi.org/10.1162/coli_a_00039

McDonald, Ryan and Giorgio Satta. 2007. On the complexity of non-projective data-driven dependency parsing. In *Proceedings of the Tenth International Conference on Parsing Technologies*, pages 121–132. https://doi.org/10.3115/1621410.1621426

Mille, Simon, Alicia Burga, Gabriela Ferraro, and Leo Wanner. 2012. How does the granularity of an annotation scheme influence dependency parsing performance? In *Proceedings of COLING 2012: Posters*, pages 839–852.

Plank, B. and G. J. M. van Noord. 2010. Dutch dependency parser performance across domains. *LOT Occasional Series*, 16:123–138.

Pretkalnina, L. and L. Rituma. 2014. Constructions in Latvian treebank: The impact of annotation decisions on the dependency parsing performance. *Human Language Technologies - The Baltic Perspective*, volume 268. IOS Press, pages 219–226.

Rehbein, Ines, Julius Steen, Bich-Ngoc Do, and Anette Frank. 2017. Universal Dependencies are hard to parse—or are they? In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 218–228.

Sagae, Kenji, Yusuke Miyao, Rune Saetre, and Jun'ichi Tsujii. 2008. Evaluating the effects of treebank size in a practical application for parsing. *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 14–20. https://doi.org/10.3115/1622110.1622114

Sen, P. K., Richard H. Lindeman, Peter F. Merenda, and Ruth Z. Gold. 1981. Introduction to bivariate and multivariate analysis. *Journal of the American Statistical Association*, 76(375):752. https://doi.org/10.2307/2287559

Shannon, C. E. 1948. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Shapiro, S. S. and M. B. Wilk. 1965. An analysis of variance test for normality (complete samples). *Biometrika*, 52:591–611. https://doi.org/10.2307/2333709, https://doi.org/10.1093/biomet/52.3-4.591

Søgaard, Anders. 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. https://doi.org/10.18653/v1/2020.emnlp-main.220

Søgaard, Anders, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. 2021. We need to talk about random splits. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1823–1832. https://doi.org/10.18653/v1/2021.eacl-main.156

Sprugnoli, Rachele, Marco Passarotti, Flavio Massimiliano Cecchini, and Matteo Pellegrini. 2020. Overview of the EvaLatin 2020 evaluation campaign. In *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 105–110.

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

Straka, Milan, Jan Hajič, and Jana Straková. 2016. UDPipe: 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.

Straka, Milan, Jan Hajič, Jana Straková, and Jan Hajič Jr. 2015. Parsing universal dependency treebanks using neural networks and search-based oracle. In *Proceedings of Fourteenth International Workshop on Treebanks and Linguistic Theories (TLT 14)*, pages 208–220.

Straka, Milan and Jana Straková. 2017. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99. https://doi.org/10.18653/v1/K17-3009

Strzyz, Michalina, 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. https://doi.org/10.18653/v1/N19-1077

Temperley, David and Daniel Gildea. 2018. Minimizing syntactic dependency lengths: Typological/cognitive universal? *Annual Review of Linguistics*, 4:67–80. https://doi.org/10.1146/annurev-linguistics-011817-045617

Timm, U. 1969. Coherent bremsstrahlung of electrons in crystals. *Protein Science*, 17(12):765–808. https://doi.org/10.1002/prop.19690171202

Vallat, Raphael. 2018. Pingouin: Statistics in Python. *Journal of Open Source Software*, 3(31):1026. https://doi.org/10.21105/joss.01026

Vaserstein, Leonid Nisonovich. 1969. Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredachi Informatsii*, 5(3):64–72.

Yap, Bee Wah and C. H. Sim. 2011. Comparisons of various types of normality tests. *Journal of Statistical Computation and Simulation*, 81(12):2141–2155. https://doi.org/10.1080/00949655.2010.520163

Zeman, Daniel, 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.

Zeman, Daniel, Martin Popel, Milan Straka, Jan Hajic, et al. 2017. Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19.

Zhang, Yi and Rui Wang. 2009. Correlating natural language parser performance with statistical measures of the text. In *KI'09: Proceedings of the 32nd Annual German Conference on Advances in Artificial Intelligence*, pages 217–224. https://doi.org/10.1007/978-3-642-04617-9_28