Investigating UD Treebanks via Dataset Difficulty Measures

Artur Kulmizev **Joakim Nivre**
Department of Linguistics and Philology
Uppsala University
{artur.kulmizev,joakim.nivre}@lingfil.uu.se

Abstract

Treebanks annotated with Universal Dependencies (UD) are currently available for over 100 languages and are widely utilized by the community. However, their inherent quality characteristics are hard to measure and are only partially reflected in parser evaluations via accuracy metrics like LAS. In this study, we analyze a large subset of the UD treebanks using three recently proposed accuracy-free dataset analysis methods: dataset cartography, $\mathcal{V}$-information, and minimum description length. Each method provides insights about UD treebanks that would remain undetected if only LAS was considered. Specifically, we identify a number of treebanks that, despite yielding high LAS, contain very little information that is usable by a parser to surpass what can be achieved by simple heuristics. Furthermore, we make note of several treebanks that score consistently low across numerous metrics, indicating a high degree of noise or annotation inconsistency present therein.

1 Introduction

Datasets have long played a crucial role in dictating the pace of progress in NLP. Their function, for most tasks, is largely two-fold: 1) to collect data points (and their corresponding gold-standard labels) on which statistical models can be trained, and 2) to serve as benchmarks though which various models can be evaluated and compared. In recent years, much research has been devoted towards developing new datasets, tasks, and benchmarks for NLP — so as to articulate the distinguishing aspects of a bevy of new neural models. Syntactic parsing has remained an active area of research in this regard, and Universal Dependencies (UD) (Nivre et al., 2016, 2020; de Marneffe et al., 2021) has emerged as a crucial initiative within NLP, offering a set of cross-lingually consistent annotation principles that have since been adapted to over 100 languages.

Notably, the CoNLL shared tasks of 2017 and 2018 (Zeman et al., 2017, 2018) featured UD at the forefront, inviting researchers to submit systems that could not only parse, but process the entirety of UD across its numerous annotation layers. Beyond parsing, UD has also been utilized for a variety of other ends, including cross-lingual transfer (Ammar et al., 2016; Pires et al., 2019; Wu and Dredze, 2020; Lauscher et al., 2020), domain adaptation (Li et al., 2019, 2020; Stymne, 2020), and linguistic typology (Futrell et al., 2015; Hahn et al., 2020; Levshina, 2019).

Though UD and other initiatives have aided in driving recent advances in NLP, overall progress has typically been measured via aggregate accuracy metrics, which provide little more than a bird's eye view into the data. In the era of deep learning, where popular models are notoriously opaque, it has thus proven vital to study the contents of datasets and identify aspects that may misrepresent model performance. In this vein, numerous studies have shown that the crowd-funded nature of some popular NLP datasets makes them prone to annotation artefacts that are readily exploitable by neural models as heuristics (Kaushik and Lipton, 2018; Gururangan et al., 2018; Poliak et al., 2018; McCoy et al., 2019). With such insights in mind, researchers have shifted their focus towards the *datasets* instead of the models, proposing general methods for exploring the former so as to better understand the performance of the latter. Such approaches have drawn from, e.g., information theory (Perez et al., 2021; Ethayarajh et al., 2022), item response theory (Rodriguez et al., 2021; Vania et al., 2021), and model training dynamics (Swayamdipta et al., 2020). This work, however, has predominately focused on classification tasks and has proven difficult to extend to other classes of problems, such as the structured prediction tasks of UD.

In this paper, we perform an analysis of (a large

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 1076–1089
May 2-6, 2023

subset of) UD v2.9 through the perspective of a popular parsing architecture — namely that of Dozat and Manning (2016). As opposed to much previous work, which prioritizes metrics like LAS in order to build accurate parsers, we aim instead to better understand the underlying data, as well as how our parser interfaces with it. To do so, we extend recently proposed dataset analysis methods based on model training dynamics (Swayamdipta et al., 2020), $\mathcal{V}$-information (Xu et al., 2020; Ethayarajh et al., 2022), and minimum description length (Blier and Ollivier, 2018; Voita and Titov, 2020; Perez et al., 2021) to the dependency parsing scenario. In working with each method, we formalize the following set of research questions:

1. Which treebanks appear *hard* (or *easy*) to parse, given a model's confidence throughout training, and variability therein?
2. Which treebanks contain the most (or least) information that is actually usable by a parser, with respect to a naive baseline?
3. Which treebanks are the most (or least) sample efficient, i.e. most easily fit by a parser, irrespective of training set size?

2 Universal Dependencies

Universal Dependencies (UD) (Nivre et al., 2016, 2020; de Marneffe et al., 2021) is an initiative focused on the development of dependency treebanks. UD is founded upon a *lexicalist* perspective on syntax, which posits that (syntactic) relations are formed directly between words. Although this approach does not take morphological segmentation explicitly into account, UD nonetheless provides such information in the form of lemmas, part-of-speech tags, and morphological features for each word. These design decisions have inspired the widespread adoption of UD as an annotation scheme, which has grown from 10 treebanks across 10 languages in v1.0 to 243 treebanks across 138 languages in v2.11.

2.1 Dependency Parsing

Though UD contains multiple layers, our focus in this paper is on its syntactic layer — the dependency tree annotation upon which parsers are trained and evaluated. As a task, dependency parsing amounts to mapping a sentence $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ to its respective syntactic structure $\mathbf{y}$. This is typically a tree (a rooted, directed, acyclic graph) over $\mathbf{x}$, where each word $x_i \in \mathbf{x}$ (called a dependent) forms an edge with another word $x_j \in \mathbf{x}$ that it syntactically modifies (its head). Though such edges (x_i, x_j) are sometimes considered in isolation, most often they are accompanied by a label describing the relation between x_i and x_j. UD comprises of a base set of 37 such labels, in addition to treebank-specific subtypes that can be introduced by annotators.

Unlike classification or sequence labeling tasks, which entail predicting y given a fixed label set K, parsing is considered a structured prediction task, where the output space is constrained by the sentence length $n + 1$ (with a special `root` symbol included). In data-driven parsing, a parser is typically a function f whose parameters θ are fit on some gold-annotated training set X_{train}, e.g. a UD treebank. A trained parser's predictions $\hat{\mathbf{y}} = f_\theta(\mathbf{x})$ can then be evaluated on a held out test set X_{test} by means of various metrics. Commonly, these are *labeled attachment score* (LAS), which is the percentage of all words in X_{test} that are assigned the correct head *and* label by f, and the *unlabeled attachment scores* (UAS), which is the percentage of words that are assigned the correct head, irrespective of label. We will focus primarily on LAS in this study, as it assesses both head attachment and labeling.

2.2 UD Parsing

In conjunction with UD's popularity and transition to v2.0, the CoNLL 2017 and 2018 Shared Tasks on Multilingual Parsing from Raw Text to Universal Dependencies (Zeman et al., 2017, 2018) sought to consolidate trends in parsing research with respect to a wealth of new data, algorithms, and models. Overall, the findings of both shared tasks offered many insights for the future of dependency parsing. Primarily, they helped establish Dozat and Manning (2016)'s parser as the most popular parsing architecture of the neural NLP era. Though many new architectures have been proposed since, the biaffine attention decoder continues to feature prominently in parsing research, where the focus has shifted to the matter of feature representation and fine-tuning, rather than decoding (see, e.g. Kondratyuk and Straka (2019); Üstün et al. (2020)).

3 Beyond Accuracy

In NLP, system performance is typically measured via accuracy, which has the advantage of being intuitive and straightforward to calculate. More-

over, it is often useful in model selection, as well in setting the state-of-the-art for a given dataset or task. Beyond these qualities, however, accuracy leaves much to be desired when investigating models vis-a-vis datasets. For one, it is often reported with respect to a single checkpoint from a model's training regime, which typically consists of numerous epochs and parameter updates. In honing in on one particular checkpoint (usually the best with respect to validation loss or accuracy), one cannot readily assess whether the model was easily fit on the data, or if training was stopped prematurely. Furthermore, in choosing the $\arg\max$ over the output distribution, one inevitably loses information about it: Was the model confident in making its prediction? Or was the distribution highly entropic? Also relevant is the train/test distinction: in evaluating on the latter, one can gauge a model's ability to generalize, but generally cannot assess the goodness-of-fit on the former, nor its sample efficiency. Likewise, accuracy cannot, in principle, adequately assess the *quality* of the training data: can the model learn from all instances therein? Or does the data contain a substantial amount of noise due to, e.g. annotation inconsistencies?

With regard to dependency parsing, accuracy-based metrics like LAS carry a number of additional drawbacks. For example, if working with an arc-factored graph-based dependency parser, one must score all possible n incoming edges for a dependent $x_i \in \mathbf{x}$. If $n = 1$, then x must necessarily be attached to the dummy node and assigned the `root` label. For a treebank consisting of many such sentences (e.g. Russian Taiga), this will lead to artificially inflated accuracies for that particular relation. Another concern is the potential proliferation of functional relations that may arise in some treebanks (e.g. Japanese GSD), where parsers often yield disproportionately high accuracies (Nivre and Fang, 2017). To transcend the limitations of accuracy-based measures in dependency parsing, we consider three recently proposed dataset analysis methods as a means of exploring UD treebanks: dataset cartography (Swayamdipta et al., 2020), $\mathcal{V}$-information (Xu et al., 2020), and minimum description length (Blier and Ollivier, 2018).

3.1 Dataset Cartography

Dataset cartography (Swayamdipta et al., 2020) is a method for analyzing training datasets via the lens of model training dynamics. Put briefly, DC assumes the use of a model f trained to minimize loss on a dataset D of size N. Crucially, for a given instance $D_i = (\mathbf{x}, y^*)$, DC posits that f defines a probability distribution over labels y, such that the probability of the true label $p(y^*|\mathbf{x})$ can be tracked throughout training. Given a gradient descent-based training regime of E epochs, DC defines the notion of *confidence* (CONF) as follows:

$$\text{CONF}_i = \frac{1}{E}\sum_{e=1}^{E} p_{\theta_e}(y^*|\mathbf{x})$$

where θ_e are f's parameters following the epoch e update. Intuitively, high CONF values (e.g. ≈ 0.95) for a given training instance D_i indicate that f generally assigns high probability to y^* throughout training — i.e. that D_i is "easy-to-learn". Conversely, low CONF values (e.g. ≈ 0.05) indicate that f generally "fails" to learn from those particular instances.

As a complement to CONF, Swayamdipta et al. (2020) also introduce the *variability* (VAR) metric, which summarizes the tendency of f to waver in its assignment of $p(y^*|\mathbf{x})$ throughout training. VAR is defined as follows:

$$\text{VAR}_i = \sqrt{\frac{\sum_{e=1}^{E}(p_{\theta_e}(y^*|\mathbf{x}) - \text{CONF}_i)^2}{E}}$$

In essence, while CONF is the mean of $p(y^*|\mathbf{x})$ across E, VAR is its standard deviation. Using both of these metrics, Swayamdipta et al. (2020) are able to construct *data maps*, which visualize D through the perspective of f. With VAR and CONF plotted on the x and y axes, respectively, data maps help in identifying select regions of D that are easy or difficult for f to learn — or are otherwise ambiguous.

3.2 $\mathcal{V}$-Information

An alternative approach for quantifying dataset "difficulty" is proposed by Ethayarajh et al. (2022), who leverage the concept of $\mathcal{V}$-Information. Introduced by Xu et al. (2020), $\mathcal{V}$-information (denoted as $I_{\mathcal{V}}(X \to Y)$) is a framework for estimating the amount of information between random variables X and Y (input and output, respectively) that is *usable* by a model in family $\mathcal{V}$ — e.g., a sentiment classifier or syntactic parser. Here, *usability* is measured with respect to an encrypted form of the input $\varnothing$, from which $\mathcal{V}$ must nonetheless attempt

to predict Y — essentially a label-only baseline. Predicting Y from X and $\varnothing$ amounts to measuring $\mathcal{V}$-entropy:

$$H_{\mathcal{V}}(Y) = \inf_{f \in \mathcal{V}} [-\log f'[\varnothing](Y)]$$

and *conditional* $\mathcal{V}$-entropy:

$$H_{\mathcal{V}}(Y|X) = \inf_{f \in \mathcal{V}} [-\log f[X](Y)]$$

where $f \in \mathcal{V}$ is a model that maximizes the log-likelihood of the labels Y with the original input X and f' is the same model trained on the encrypted input $\varnothing$. Given these two quantities, $I_{\mathcal{V}}(X \to Y)$ can be computed as follows:

$$I_{\mathcal{V}}(X \to Y) = H_{\mathcal{V}}(Y) - H_{\mathcal{V}}(Y|X)$$

In essence, $\mathcal{V}$-information is the amount of information between inputs and labels that can be estimated by $\mathcal{V}$ beyond the label distribution itself. Given that $\mathcal{V}$-information is computed with respect to $H_{\mathcal{V}}(Y)$, it is important to note that $I_{\mathcal{V}}(X \to Y) \geq 0$. Also, if X is independent of Y, then $I_{\mathcal{V}}(X \to Y) = 0$.

In addition to functioning as a summary statistic alternative to accuracy, $\mathcal{V}$-information can also be generalized to the instance-case. To do so, Ethayarajh et al. (2022) propose measuring point-wise difficulty, which they deem PVI and calculate as follows:

$$\text{PVI}(x \to y) = -\log_2 p_{f'}(y^*|\varnothing) + \log_2 p_f(y^*|\mathbf{x})$$

where $f_\theta, f'_\theta \in \mathcal{V}$ are models trained on normal and encrypted data, respectively. Recall that, as before, y^* refers to the gold label and not the one with the highest score. Unlike $\mathcal{V}$-information, PVI can return negative values at the instance level (similarly to pointwise mutual information (Shannon, 1948)), which indicates that the model would fare better choosing a class at random.

3.3 Minimum Description Length

Minimum description length (MDL) (Rissanen, 1978) is an information-theoretic concept that concerns the transmission of data through a specified channel — i.e., a probabilistic model. Ideally, a model that is fit well on some data will learn to transmit — or *compress* — it using as few bits as possible (Blier and Ollivier, 2018). Naively, in order to evaluate how well a model, e.g., a neural network, might learn to compress its training data, one might refer to the model's cross-entropy loss after training for a full cycle of E epochs, which amounts to the Shannon-Humman code (Shannon, 1948). However, a model endowed with enough parameters may learn to *fit* the data without necessarily *compressing* it — see, e.g., Zhang et al. (2021), who show that training loss can still be minimized on data that contains no inherent structure (shuffled labels). MDL is thus designed to express not only how well a model might learn to compress some data, but also how efficiently the model itself might be transmitted.

Blier and Ollivier (2018) outline various methods for compressing labels, with or without a model. Among these, they describe online (or prequential) coding (Rissanen, 1984; Dawid, 1984), which transmits the labels and model without explicitly compressing the latters's parameters. Online coding requires D to be partitioned into S blocks where $1 = t_0 < t_1 \cdots < t_S = N$. The model of choice, f, is initialized with parameters θ, learning algorithm $\mathcal{A}$, etc. The first block, t_0, is first evaluated with a uniform prior[1] and then used to train f_s (we omit the parameters θ for brevity). This model is then evaluated on t_{s+1} and reset to its initial state, where it is consequently trained on t_{s+1}, and so on. Formally, the online codelength can be expressed as:

$$L^{\text{online}}(y_{1:n}|x_{1:n}) = \sum_{s=1}^{S-1} \sum_{n=t_s}^{t_{s+1}} -\log_2 p_{f_s}(y_n|x_n)$$

In contrast to the training loss above, which represents the data codelength if the model parameters are known, L^{online} (henceforth MDL) is an implicit way of measuring the same without knowing the model parameters. Effectively, MDL estimates f's ability to generalize with respect to D: models that learn efficiently from limited instances will yield shorter codelengths.

4 Experimental Setup

In this section, we describe our data sampling procedure, the parsing model we employ, and our extension of the aforementioned analysis methods to the context of (graph-based) dependency parsing. More details about how each metric is calculated can be found in Appendix A.

Data In order to compare the analysis obtained with each of the dataset analysis methods, we require a representative sample of UD treebanks.

[1] More details on this can be found in Appendix A.

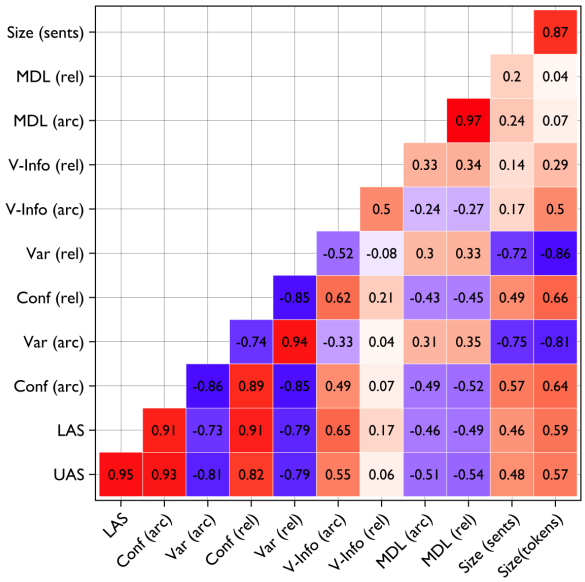

Figure 1: Spearman's ρ across all metrics of interest, including training set size in tokens and sentences.

Since each method requires the training of a parser on some subset of data (as well as a validation set for estimating V-INFO), we consider every UD treebank that contains train, validation, and test splits. To ensure that a) there is sufficient data for training our parsers and b) training is reasonably quick and the models do not overfit, we limit our selection to treebanks whose training sets contain at least 1,000 and at most 20,000 sentences. This gives us 88 treebanks across 58 languages,[2] with Faroese FarPahC having the smallest training partition (1,021 sentences, 23,094 tokens) and Polish PDB the biggest (17,773 sentences, 281,736 tokens). All resuls are reported with respect to UD v2.9, which was the most recent release at the time of experimentation.

Parsing Model We employ a neural parser based on Dozat and Manning (2016)'s biaffine decoder, as implemented in the `SuPar`[3] Python library. For input encoding, we make use of randomly initialized word embeddings ($d \in \mathbb{R}^{100}$) and LSTM-based character embeddings ($d \in \mathbb{R}^{100}$), as well as a stacked three-layer LSTM feature extractor ($d \in \mathbb{R}^{400}$). We choose to forego the use of POS embeddings (contrary to Dozat et al. (2017)), so as to maintain a direct correspondence between the input string and its tree, which is vital for measuring V-INFO. Furthermore, we do not initialize the input embeddings with pretrained representations in order to avoid confounds central to language coverage and overlap (Wu and Dredze, 2020). Each model is trained for 30 epochs with a batch size of 32 and optimized by Adam (Kingma and Ba, 2014) with a starting learning rate of $2e{-}3$.

Analysis Methods In order to be able to measure CONF, VAR, and V-INFO, we need to be able to extract the probabilities that parsers assign to gold arcs and labels. Recall that Dozat and Manning (2016)'s parser is effectively a multi-task model, which jointly maximizes the log-likelihood of a given word's correct head, as well as the label for the relation. The arc and label logits are calculated via separate biaffine transformations, which yield a $S_{\text{arcs}} \in \mathbb{R}^{N \times N+1}$ matrix for the former and a $S_{\text{labels}} \in \mathbb{R}^{N \times N+1 \times R}$ tensor for the latter, where N is the sentence length and R is the size of the relation set. To obtain a normalized probability distribution, we apply a softmax to the last dimension of each S_{arcs} and S_{labels}, and index into the correct cell for the gold head and label probabilities.

For MDL, we train parsers on increasingly larger partitions of the training set, starting with a minimum of five sentences and doubling in size up to a maximum of 360 — a total of 995 sentences. We do so in reference to the smallest treebank in our sample (Faroese FarPahC (1,021 sentences)), so as to control for training set size, which varies drastically across treebanks. In accordance with the parser's multi-task design, we compute two separate MDL measures with respect to the separate arc and relation losses, averaged over five trials.

5 Results and Analysis

In this section, we analyze the results of our four metrics: CONF, VAR, V-INFO, MDL. We focus our analysis on arcs (probabilities and loss) and refer to Appendix B for the full set of results. For each metric, we begin with a brief discussion of its pairwise correlation to all other metrics, which is displayed for reference in Figure 1. We continue by framing our analyses with respect to the Top 3 best and worst performing treebanks for each metric.

5.1 Dataset Cartography

Figure 2 (left) depicts the mean CONF and VAR scores calculated across arcs. We observe a strong negative correlation between CONF and VAR. This relationship also appears to be influenced by 1) validation LAS and 2) training set size. This is not surprising, as training set size is a common

[2] We filter out four treebanks due to issues with tokenization, etc.

[3] https://github.com/yzhangcs/parser

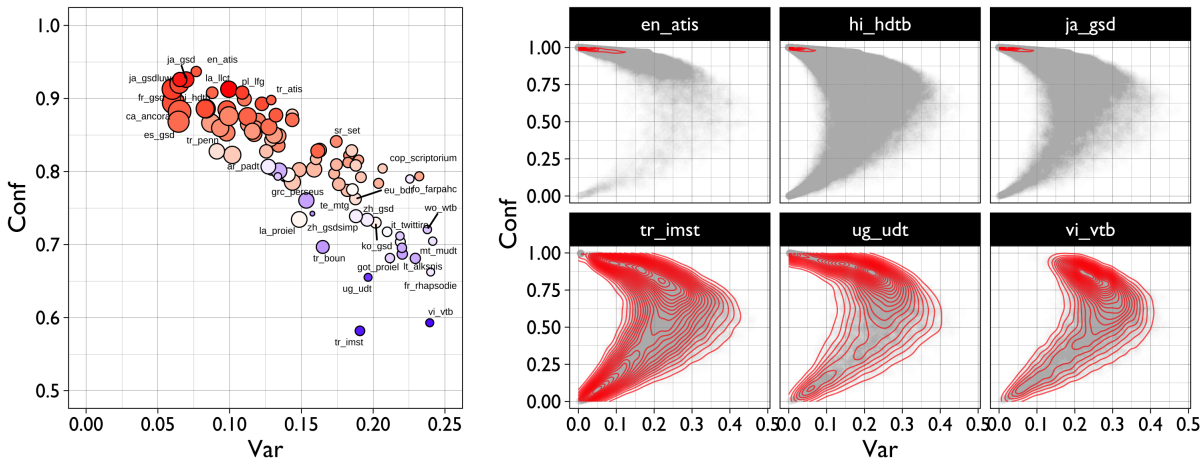

Figure 2: Left: mean CONF and VAR scores for Arcs across all languages; color represents Validation LAS and point size represents the size of the train set in words. Right: Data maps for Arcs for best and worst CONF scoring languages, with 2d densities super-imposed.

predictor of indicates that parser performance can, to a large extent, be reliably estimated simply by observing its confidence throughout training.

In zooming in on individual points, we observe several treebanks in the upper left-hand corner of the arc plot, which corresponds to high average CONF and low VAR. Indeed, many of these points belong to the largest treebanks in the sample, e.g., Hindi HDTB (0.92 CONF, 0.06 VAR, 281,057 tokens), French GSD (0.92, 0.06, 354,505), and Catalan AnCora (0.89, 0.06, 429,141). However, English Atis — a relatively small treebank with 48,655 tokens— tops out with the highest CONF overall at 0.94 (VAR: 0.08). This is unsurprising given the nature of the Atis dataset (Price, 1990), which collects transcriptions of requests sent to automated flight information systems, e.g. *list the nonstop flights early tuesday morning from dallas to atlanta*. The imperative nature of such requests, in combination with a small vocabulary, likely limits the range of structures a parser might encounter during training, thus making the treebank *easy* to fit. Interestingly, the second and third-highest CONF treebanks are both Japanese: GSDLUW (0.93, 0.07, 130,298) and GSD (0.93, 0.07, 168,333). This is likewise expected, as GSD has been observed by, e.g., Nivre and Fang (2017) to possess a large amount of functional relations, which can be parsed with near 100% accuracy.

On the other end of the spectrum, we observe that Turkish IMST (0.58, 0.19, 37,784), Vietnamese VTB (0.59, 0.24, 20,285), and Uyghur UDT (0.67, 0.20, 19,262) yield the three lowest CONF scores overall, as well as generally high VAR. Interestingly, though Turkish IMST does not appear within the top-25 highest VAR treebanks, it nonetheless yields the lowest CONF, indicating the the treebank might be particularly "hard" to parse for reasons other than treebank size (Çöltekin et al., 2017). Conversely, the fact that VTB and UDT are low CONF and high VAR implies that a lack of training data might play a role.

The data maps for the highest and lowest CONF scoring treebanks (depicted in Figure 2 (right)) highlight important disparities between these two groups. Most strikingly, we observe that the overwhelming majority of arcs in the English, Hindi, and Japanese treebanks are concentrated in the upper left high-CONF/low-VAR region, which is characterized as easy-to-learn by Swayamdipta et al. (2020). Indeed, this corresponds to 65.26, 68.02, and 68.32% of all tokens in the English, Hindi, and Japanese treebanks where CONF ≥ 0.95 and VAR ≤ 0.1. Conversely, only 3.04, 7.61, and 0.001% of all arcs in the Turkish, Uyghur, and Vietnamese treebanks are allocated to this region, indicating that this group contains few arcs that might be considered "trivially easy". Likewise, if we define the hard-to-learn region as consisting of points where CONF ≤ 0.25 and VAR ≤ 0.1, we find that 9.5, 5.45, and 5.06% of the latter group's treebanks can be characterized as such, respectively, compared to 0% for any treebank in the former group.

5.2 $\mathcal{V}$-Information

Figure 3 depicts the PVI density for the Top 3 and Bottom 3 treebanks in terms of V-INFO for arcs.

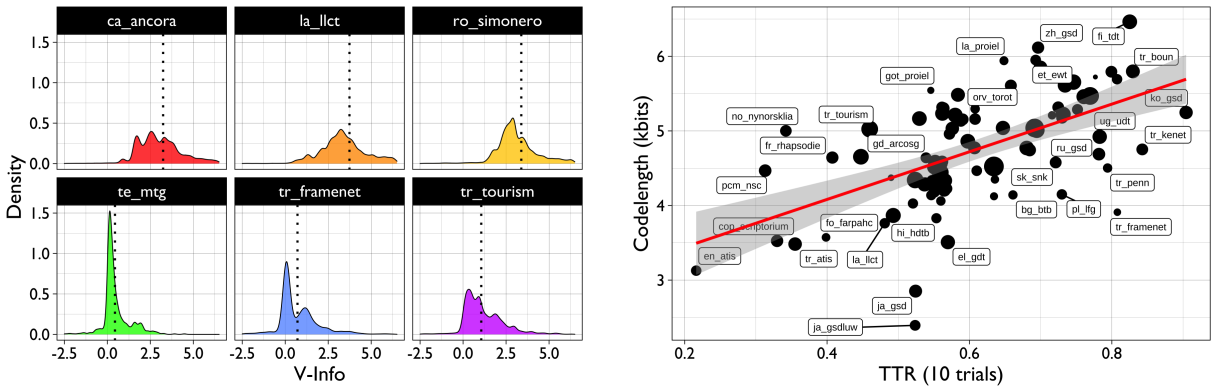

Figure 3: Left: Arc-level PVI density for Top-3 and Bottom-3 V-INFO treebanks, across arcs (labels omitted for space). Right: Block-wise codelength (in bits) for Top-3 and Bottom-3 MDL treebanks, across arcs.

We observe weaker correlations between V-INFO and LAS when measured across all treebanks than we did for CONF or VAR, indicating that V-INFO measures different aspects of parser performance than either of these metrics. Immediately, we can see that the lowest-ranked treebanks — Telugu MTG (0.44 average V-INFO), Turkish FrameNet (0.71), and Turkish Tourism (1.05) — have densities that are skewed towards 0. In particular, Telugu shows a sharp peak around this point, suggesting that it contains many arcs and relations that can be inferred by sentence sizes and label distributions. In contrast, the distributions representing the highest V-INFO treebanks (Latin LLCT (3.74), Romanian SiMoNERo (3.38), and Catalan AnCora (3.24)) are much flatter and more evenly distributed along the space of positive V-INFO values.

Turkish Tourism is composed entirely of hotel and restaurant reviews. Due to the nature of this genre, the treebank's vocabulary is very limited and many of its sentences are exceptionally short — 4.77 tokens on average in the dev set. This limits the space of possible trees that a parser may encounter, potentially biasing it towards certain structures. For example, the adverb "çok" (very) appears in the first position of 128 sentences as modifier of the 2nd word (typically an adjective, e.g., "güzel" (good)). A similar effect can be observed for Telugu MTG, which contains 5.05 words per sentence in its development set, and for Turkish FrameNet, which systematically places the root of the sentence at the penultimate position (91.7% of the time).

Given the observations above, V-INFO can be imagined as a means of simultaneously penalizing regularity and stochasticity in data. We can illustrate this further by returning to the CONF results. Recall that, for arcs, the highest scoring treebank was English Atis. Interestingly, when measuring its arc V-INFO, we find that its rank drops to 77 out of 88 (1.97). Recall that the Atis dataset contains a single genre of data with simple sentences often beginning with *what is*, *show me*, etc. In this case, simply guessing $(1, 0)$ and $(2, 1)$ for the first two arcs yields accuracies of 0.52 and 0.41, respectively, which the label-only baseline employed for calculating V-INFO would likewise capture. By contrast, the lowest scoring CONF treebank — Turkish IMST — retains a low rank of 83 (1.27). Recalling that 9.5% of IMST's tokens are considered hard-to-learn by the cartography metrics, this treebank is likely to contain certain annotation inconsistencies that cannot be systematically captured beyond guessing.[4] Such cases, too, are captured by V-INFO. With this in mind, we might surmise that *high* V-INFO values might represent treebanks with a varied distribution of structures that are likewise consistently annotated, thus requiring strong generalization of the parser.

In addition to arcs, we would also like to note the interesting behavior of V-INFO for relations, which is generally uncorrelated with any other metric (other than arcs). This is due to the fact that the parser must know the identity of both words when deciding upon the relation that binds them. For the vast majority of relations, this cannot be determined from position alone, even if head placement is largely systematic (as in the aforementioned Atis and Tourism treebanks). As a result, aggregate

[4]Indeed, this is corroborated by Türk et al. (2019), who discuss the errors resulting from the automatic conversion of IMST to the UD format, among which are 269 cases where a subject bearing a genitive marker was misidentified as an object in the accusative case.

V-INFO for relations (3.89 ± 3.66) is noticeably higher than for arcs (2.40 ± 1.98), albeit with more spread. It is important to note, however, that there are two primary exceptions to this in the form of the `root` and `punct` relations, which can be deterministically assigned to the first (dummy node) and last (end of sentence punctuations) positions.

5.3 MDL

In terms of MDL, we observe only moderate correlations across all other measures, with the strongest between CONF for arcs. Here, we find that MDL for relations (which takes the model's loss into account) is actually more correlated with arc CONF than CONF for relations. Expectably, neither measure is correlated with treebank size, which was controlled across treebanks.

For rankings, we observe that the Top 3 treebanks (for arcs) are largely the same ones as returned by arc CONF: Japanese GSD and GSDLUW, (2.85, 2.39) English Atis (3.12), and Turkish Atis (3.48). This is unsurprising given the reasons outlined in previous sections. Therefore, it seems feasible that MDL — which measures a parser's fit on successively larger, unseen partitions of a dataset — would reflect such qualities as well. MDL's alternative interpretation as a measure of a dataset's sample efficiency is also applicable here: a parser trained on a small number of trees will likely generalize well in these very narrow distributions.

Interestingly, we encounter a handful of new treebanks at the higher end of MDL scores: Finnish TDT (6.46), Chinese GSD (traditional and simplified) (6.11, 6.14), and Latin PROIEL (5.94). Intuitively, a high MDL in the case of parsing might suggest that the model is exposed to a larger diversity of token types during training, which could hinder it in learning various types of dependencies. Following this logic, one might hypothesize that treebanks representing languages with complex morphological systems might yield comparatively higher MDL, due to the higher number of word forms that appear therein. In line with this, we observe that the highest MDL treebanks tend to be morphologically rich, e.g., Finnish, Latin, Turkish, Estonian, Polish, Russian.

In an attempt to quantify the correspondence between a treebank's attested morphological complexity and its MDL, we compute a series of proxy metrics, as featured in Çöltekin and Rama (2018). These include type-token ratio (TTR) (averaged across 10 random samples of 1,000 sentences), number of feature types (in these samples), and feature entropy (calculated across feature types). Indeed, Figure 3 shows a strong correspondence between TTR and MDL, as the two are highly correlated ($\rho = 0.58, p < 0.001$). Significant correlations for feature entropy ($\rho = 0.35, p < 0.001$) and number of feature types ($\rho = 0.28, p < 0.001$) corroborate our hypotheses further. Interestingly, while both Chinese GSD treebanks simultaneously yield the highest MDL and comparatively high morphological scores, the language itself is typically described as having an analytical (impoverished) morphology. We surmise, however, that this fact — combined with Chinese's logographic writing system — contributes to high MDL scores in the same way as morphological richness: a lack of high-frequency function words and a wide range of lexical items lead to large vocabulary sizes. MDL for these treebanks is thus expectably high.

6 Conclusion

In this paper, we investigated 88 UD treebanks through the lens of dataset difficulty measures. We found that CONF and VAR are capable of painting a nuanced picture of how *easy* or *hard* treebanks might be to parse. We also observed that a model's confidence throughout training is an excellent indicator of how well it might generalize to held-out data. Regarding V-INFO, we observed that the measure tends to simultaneously penalize high degrees of predictability and stochasticity, and that treebanks otherwise characterized as *easy* may have low V-INFO due to lack of structural diversity. Finally, treebanks with high MDL seem to be characterized by low sample efficiency, which in turn is related to morphological complexity and vocabulary usage. Given the broad range of insights expressed via these metrics, we hope that our results — however preliminary — will inspire future researchers to pursue a greater understanding of UD as trove of data, so as to push further boundaries in the realms of parsing, typology, etc.

In terms of future work, we make note of several potentially interesting directions. Indeed, one of the main drawbacks of our experimental design is that it only accounts for the perspective of a single parsing architecture — albeit (arguably) the most popular one of the neural era. Dependency parsing, however, has a long research tradition where many different parsing models have been proposed

throughout the years — each with their respective advantages and drawbacks (see, e.g. McDonald and Nivre (2007, 2011)). Though we chose to work with an arc-factored graph-based parser due to the need for extracting arc-level probabilities, future studies may consider consider ways of leveraging transition-based parsers (Nivre, 2003) or models that directly maximize full tree probabilities (Koo et al., 2007; Ma and Hovy, 2017). If working with a wide range of parsing models, one could employ item response theory (Rodriguez et al., 2021; Vania et al., 2021), which is a framework that consolidates many predictions per instance in order to identify regions of datasets that may be perceived as difficult, easy, etc. Certainly, this would provide a more broad perspective on UD than what we have offered here.

A different direction that could be explored is data *selection*, which is indeed what Swayamdipta et al. (2020) proposed as the main uses for dataset cartography. Although the CONF and VAR metrics provide valuable insights about UD treebanks in our case, they are nonetheless measured at the *token*-level. This is distinct from their original application, in that each token is crucial for the composition a sentence and cannot be readily removed. Although we experimented with measuring sentence-level CONF and VAR, some preliminary results indicated that a naive application of Koo et al. (2007)'s method is ultimately confounded by sentence length. As such, it would be interesting to experiment with models that directly optimize for tree probability, such as Ma and Hovy (2017). If successful, this would allow us to identify select subsets of treebanks for the purpose of training more accurate parsers with less data, or for choosing the least noisy sentences for typological studies, etc.

Limitations

As already mentioned, the main limitation to our work is that we focus on a single parsing architecture. Indeed, it would be preferable to extend the experiments described here to other parsers in order to evaluate the generalizability of our results. Ideally, we might choose to work with a comparable transition-based parsing algorithm, which have been shown to exhibit different error profiles than their graph-based counterparts (McDonald and Nivre, 2007; Kulmizev et al., 2019). However, the fact that transition-based parsers calculate probabilities over *transitions* instead of *arcs* would render such parsers incompatible with the cartography and V-INFO measures, which reveal interesting insights about our surveyed treebanks. Beyond this, we note that the scope of our current work is quite large, as we compare 10 metrics across 88 treebanks. By this token, we can admittedly only offer a bird's eye view of the UD treebank collection, even if our surveyed metrics offer more nuance than attachment scores.

Ethical Considerations

The research presented in this paper is compatible with the ACL ethics policy. The datasets used come from the Universal Dependencies repository and have appropriate licenses and documentation. The experiments are done with small-scale models that do not have a significant impact in terms of energy consumption.

Acknowledgements

We would like to thank Anders Søgaard for providing useful feedback on an early version of this work, as well as Sasha Berdicevskis for lending insight about computational measures of morphological complexity. We acknowledge the computational resources provided by CSC in Helsinki through NeIC-NLPL (`www.nlpl.eu`).

References

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

Léonard Blier and Yann Ollivier. 2018. The description length of deep learning models. *arXiv preprint arXiv:1802.07044*.

Çağrı Çöltekin, Ben Campbell, Erhard Hinrichs, and Heike Telljohann. 2017. Converting the tüba-d/z treebank of german to universal dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 27–37.

Çağrı Çöltekin and Taraka Rama. 2018. Exploiting universal dependencies treebanks for measuring morphosyntactic complexity. *Proceedings of the First Shared Task on Measuring Language Complexity*, pages 1–8.

A Philip Dawid. 1984. Present position and potential developments: Some personal views statistical theory the prequential approach. *Journal of the Royal Statistical Society: Series A (General)*, 147(2):278–290.

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

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. pages 5988–6008.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Michael Hahn, Dan Jurafsky, and Richard Futrell. 2020. Universals of word order reflect optimization of grammars for efficient communication. *Proceedings of the National Academy of Sciences*, 117(5):2347–2353.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, Brussels, Belgium. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. 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 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Terry Koo, Amir Globerson, Xavier Carreras Pérez, and Michael Collins. 2007. Structured prediction models via the matrix-tree theorem. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 141–150.

Artur Kulmizev, 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, Hong Kong, China. Association for Computational Linguistics.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Natalia Levshina. 2019. Token-based typology and word order entropy: A study based on universal dependencies. *Linguistic Typology*, 23(3):533–572.

Ying Li, Zhenghua Li, and Min Zhang. 2020. Semi-supervised domain adaptation for dependency parsing via improved contextualized word representations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3806–3817, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zhenghua Li, Xue Peng, Min Zhang, Rui Wang, and Luo Si. 2019. Semi-supervised domain adaptation for dependency parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2386–2395, Florence, Italy. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2017. Neural probabilistic model for non-projective MST parsing. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 59–69, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 122–131, Prague, Czech Republic. Association for Computational Linguistics.

Ryan McDonald and Joakim Nivre. 2011. Analyzing and integrating dependency parsers. *Computational Linguistics*, 37(1):197–230.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 149–160, Nancy, France.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Joakim Nivre and Chiao-Ting Fang. 2017. Universal Dependency evaluation. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 86–95, Gothenburg, Sweden. Association for Computational Linguistics.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. Rissanen data analysis: Examining dataset characteristics via description length. *arXiv preprint arXiv:2103.03872*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Patti Price. 1990. Evaluation of spoken language systems: The atis domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Jorma Rissanen. 1978. Modeling by shortest data description. *Automatica*, 14(5):465–471.

Jorma Rissanen. 1984. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information theory*, 30(4):629–636.

Pedro Rodriguez, Joe Barrow, Alexander Miserlis Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. 2021. Evaluation examples are not equally informative: How should that change NLP leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4486–4503, Online. Association for Computational Linguistics.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Sara Stymne. 2020. Cross-lingual domain adaptation for dependency parsing. In *Proceedings of the 19th International Workshop on Treebanks and Linguistic Theories*, pages 62–69, Düsseldorf, Germany. Association for Computational Linguistics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Utku Türk, Furkan Atmaca, Şaziye Betül Özateş, Balkız Öztürk Başaran, Tunga Güngör, and Arzucan Özgür. 2019. Improving the annotations in the turkish universal dependency treebank. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 108–115.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly Universal Dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315, Online. Association for Computational Linguistics.

Clara Vania, Phu Mon Htut, William Huang, Dhara Mungra, Richard Yuanzhe Pang, Jason Phang, Haokun Liu, Kyunghyun Cho, and Samuel R. Bowman. 2021. Comparing test sets with item response theory. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1141–1158, Online. Association for Computational Linguistics.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings*

of the 5th Workshop on Representation Learning for NLP, pages 120–130, Online. Association for Computational Linguistics.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. 2020. A theory of usable information under computational constraints. *arXiv preprint arXiv:2002.10689*.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

A Analysis Method Details

Since every surveyed dataset analysis method was proposed with classification tasks in mind, we must make numerous modifications in order to make them applicable to structured prediction problems like dependency parsing.

Dataset Cartography Given that we can index directly into S_{arcs} and S_{labels} throughout the training regime, the process of calculating CONF and VAR for each token is relatively straightforward. As such, we keep track of the probabilities assigned to the gold arcs and labels in the train and development sets after each epoch.

$\mathcal{V}$-information In order to calculate V-INFO, we must be able to estimate $\mathcal{V}$-entropy and conditional $\mathcal{V}$-entropy. Though the latter can be computed by simply fitting a model $f' \in \mathcal{V}$ on a designated training set, the former requires the "encryption" of the input X. To do so, we follow Ethayarajh et al. (2022) in setting every input token string $x \in X$ to _. $\mathcal{V}$-entropy can then be estimated by fitting f' on the same, albeit encrypted, training set, and $I_{\mathcal{V}}(X \to Y)$ subsequently calculated on a held-out (also encrypted) validation set. Though we track V-INFO across all epochs, we report it for $e = 30$, across arcs and labels. Also, it is worth noting that, although we attempted to compute $I_{\mathcal{V}}(X \to Y)$ at the sentence level via unlabeled tree probabilities extracted via Koo et al. (2007)'s method, the model f' trained on encrypted data produced extremely low probabilities, which led to underflow when computing the logarithm. As such, we chose to forego further exploration of this problem for this study.

Minimum Description Length Since our treebanks vary greatly in size, we must set our partitions such that they can span the length of the smallest training set (1,021 sentences). This way, our estimation of MDL remains comparable across treebanks. To do so, we employ partitions $S = \{5, 10, 20, 40, 80, 160, 320, 360\}$ at the sentence level, where the entire training set is shuffled prior to partitioning. For t_0, which does not contain any training data, we follow Voita and Titov (2020) in calculating the codelength over t_0 using a uniform prior. This is computed as follows:

$$L^{\text{unif}}(y_{1:s_1}|x_{1:s_1}) = \sum_{i=1}^{n} \log_2 K_{t_{1_i}} \quad (1)$$

where K is the number of words (arcs or labels) in a given sentence i in the first partition t_1 (composed of $s = 5$ sentences). Note that this is the extension of uniform coding proposed by Blier and Ollivier (2018) to structured prediction. In a typical classification task with K labels, the same codelength is computed as $L^{\text{unif}}(y_{1:s_1}|x_{1:s_1}) = s_1 \log_2 K$.

For all remaining $t \in S$, we revert a model to its initial state, train it on t_i and compute its codelength over t_{i+1}. Since MDL is particularly sensitive to the ordering of instances within S, we repeat this process for 5 trials, fully re-initializing the model after each one.

B Full Results

The results for all metrics on all treebanks can be found in Table 1.

		Size	Rank	Accuracy		Arcs				Relations			
Language	Treebank	# Tokens	Abs. Rank	UAS	LAS	CONF	VAR	V-INFO	MDL	CONF	VAR	V-INFO	MDL
Afrikaans	AfriBooms	33.88	46	84.48	79.48	0.80	0.21	2.72	3.87	0.89	0.13	2.64	3.69
Ancient Greek	PROIEL	187.03	55	85.00	79.92	0.79	0.14	2.83	5.30	0.87	0.09	2.66	5.10
Ancient Greek	Perseus	159.90	75	75.13	68.61	0.76	0.15	2.57	5.32	0.85	0.10	2.57	5.31
Arabic	PADT	223.88	48	85.45	79.39	0.82	0.10	2.43	4.52	0.88	0.06	2.31	4.94
Armenian	ArmTDP	42.10	66	83.07	74.69	0.70	0.22	2.97	5.18	0.76	0.17	2.90	5.27
Basque	BDT	72.97	71	82.44	77.90	0.76	0.19	2.34	5.29	0.87	0.13	2.48	5.21
Bulgarian	BTB	124.34	15	91.61	86.76	0.90	0.11	2.83	4.14	0.93	0.07	2.60	4.37
Catalan	AnCora	429.14	2	92.98	90.13	0.89	0.06	3.24	4.34	0.94	0.03	2.68	4.17
Chinese	GSD	98.62	77	78.68	73.78	0.73	0.20	2.78	6.12	0.87	0.15	2.82	6.14
Chinese	GSDSimp	98.62	74	78.76	73.90	0.74	0.19	2.78	5.95	0.87	0.13	2.82	6.20
Coptic	Scriptorium	30.84	54	86.36	82.54	0.79	0.23	2.86	3.53	0.86	0.21	2.43	3.74
Croatian	SET	152.86	31	88.44	83.86	0.85	0.13	3.02	5.21	0.90	0.08	2.81	5.00
Czech	FicTree	133.64	24	91.28	86.52	0.87	0.12	2.90	4.77	0.89	0.09	2.84	4.74
Danish	DDT	80.38	53	83.88	80.56	0.77	0.18	2.78	5.13	0.90	0.10	2.87	5.15
Dutch	Alpino	186.03	29	91.35	87.44	0.86	0.13	3.03	5.48	0.91	0.07	2.78	5.42
Dutch	LassySmall	75.13	52	84.95	78.94	0.83	0.19	2.85	4.79	0.89	0.11	2.71	5.00
English	Atis	48.66	13	93.63	89.39	0.94	0.08	1.97	3.13	0.94	0.07	1.48	2.99
English	EWT	204.58	23	88.95	85.98	0.85	0.10	2.79	5.03	0.94	0.05	2.96	5.24
English	GUM	103.40	25	88.66	85.65	0.84	0.13	3.05	4.86	0.92	0.08	3.04	5.00
English	LinES	57.37	45	85.77	81.43	0.80	0.17	2.78	4.64	0.89	0.12	2.81	4.64
English	ParTUT	43.50	44	86.52	81.67	0.78	0.20	2.86	4.21	0.88	0.14	2.87	4.39
Estonian	EWT	49.60	86	75.94	68.64	0.69	0.22	2.31	5.85	0.78	0.17	2.60	5.73
Faroese	FarPaHC	23.09	62	79.24	73.43	0.79	0.23	2.43	3.57	0.86	0.17	2.49	3.57
Finnish	FTB	127.60	69	86.54	81.32	0.80	0.15	2.07	5.79	0.86	0.12	2.34	5.76
Finnish	TDT	162.82	59	85.68	80.96	0.80	0.16	2.68	6.46	0.88	0.12	3.09	6.42
French	GSD	354.50	4	94.49	91.46	0.91	0.06	3.13	4.44	0.94	0.04	2.62	4.45
French	Rhapsodie	19.14	78	80.63	73.45	0.66	0.24	2.57	4.64	0.79	0.19	2.58	4.62
French	Sequoia	50.51	26	89.12	85.50	0.82	0.18	3.16	4.14	0.90	0.12	2.70	4.13
Galician	CTG	79.33	37	85.13	81.83	0.79	0.14	2.82	4.03	0.91	0.07	2.15	4.23
German	GSD	263.80	19	88.23	83.32	0.87	0.09	2.79	4.35	0.94	0.04	2.77	4.52
Gothic	PROIEL	35.02	84	79.59	71.86	0.68	0.21	2.26	5.54	0.77	0.17	2.45	5.57
Greek	GDT	42.33	37	86.89	82.84	0.82	0.19	3.00	3.51	0.88	0.13	2.53	3.60
Hebrew	HTB	137.72	18	88.68	84.94	0.86	0.12	2.93	4.18	0.92	0.07	2.77	4.38
Hindi	HDTB	281.06	6	94.10	90.57	0.92	0.06	3.02	3.83	0.94	0.05	2.05	3.90
Icelandic	Modern	123.87	34	91.32	88.40	0.83	0.16	2.79	5.17	0.90	0.11	2.82	5.14
Indonesian	GSD	97.60	49	85.86	78.71	0.80	0.13	2.88	5.04	0.85	0.10	2.70	4.83
Irish	IDT	95.88	32	86.61	80.02	0.84	0.13	2.60	4.23	0.89	0.10	2.73	4.29
Italian	ISDT	276.02	12	91.78	88.80	0.89	0.08	2.98	4.59	0.94	0.05	2.61	4.41
Italian	ParTUT	48.93	40	87.63	83.48	0.81	0.18	2.88	4.28	0.90	0.11	2.58	4.42
Italian	PoSTWITA	99.54	60	83.23	76.76	0.80	0.14	2.63	5.32	0.86	0.09	2.76	5.36
Italian	TWITTIRO	23.63	76	79.60	71.49	0.71	0.22	2.56	4.63	0.78	0.17	2.49	4.62
Italian	VIT	225.08	21	87.94	84.94	0.86	0.09	3.04	4.59	0.92	0.05	2.55	4.46
Japanese	GSD	168.33	3	94.78	93.46	0.93	0.07	2.88	2.85	0.97	0.04	2.06	2.62
Japanese	GSDLUW	130.30	8	94.86	93.70	0.93	0.07	2.46	2.39	0.97	0.04	1.85	2.28
Korean	GSD	56.69	80	81.91	76.14	0.73	0.20	2.13	5.25	0.83	0.16	2.07	5.29
Latin	LLCT	194.14	1	96.45	95.12	0.91	0.10	3.74	3.76	0.96	0.05	2.90	4.02
Latin	PROIEL	172.13	67	81.81	75.85	0.73	0.15	2.49	5.94	0.83	0.11	2.87	6.04
Latvian	LVTB	201.60	41	87.03	83.32	0.85	0.13	2.83	5.72	0.92	0.09	2.99	5.57
Lithuanian	ALKSNIS	47.64	81	76.43	69.36	0.68	0.23	2.43	5.47	0.81	0.16	2.75	5.29
Maltese	MUDT	22.88	67	81.10	73.05	0.70	0.24	2.69	4.47	0.77	0.21	2.83	4.64
Naija	NSC	111.88	11	93.09	90.47	0.89	0.12	3.18	4.47	0.93	0.07	2.67	4.37
Norwegian	Bokmaal	243.89	14	91.88	89.37	0.88	0.10	2.87	5.15	0.95	0.05	2.86	5.09
Norwegian	Nynorsk	245.33	15	90.97	88.47	0.88	0.11	2.95	5.12	0.94	0.06	2.92	4.98
Norwegian	NynorskLIA	35.21	79	75.58	69.24	0.70	0.22	2.17	5.00	0.83	0.16	2.61	5.18
Old Church Slavonic	PROIEL	37.43	81	81.87	74.62	0.72	0.21	2.31	5.30	0.79	0.18	2.43	5.23
Old East Slavic	TOROT	118.63	73	81.85	75.31	0.80	0.14	2.04	5.61	0.86	0.10	2.35	5.74
Old French	SRCMF	158.62	39	89.07	83.86	0.86	0.11	2.67	5.23	0.90	0.07	2.68	5.28
Persian	Seraji	121.06	36	88.28	84.13	0.83	0.16	3.14	4.77	0.91	0.08	2.44	5.04
Polish	LFG	104.75	20	94.43	91.37	0.91	0.11	2.23	4.15	0.92	0.08	2.27	4.27
Polish	PDB	281.74	30	91.63	85.57	0.88	0.10	2.89	5.65	0.88	0.08	3.08	5.68
Portuguese	Bosque	171.78	17	90.22	86.57	0.85	0.12	3.12	4.26	0.92	0.06	2.60	4.34
Portuguese	GSD	255.31	7	92.31	90.37	0.89	0.08	3.09	4.06	0.96	0.04	2.45	4.27
Romanian	RRT	185.11	28	89.69	83.80	0.86	0.12	2.97	5.04	0.89	0.07	2.85	4.96
Romanian	SiMoNERo	116.86	9	92.43	89.19	0.88	0.13	3.38	4.12	0.93	0.07	2.69	4.18
Russian	GSD	74.91	47	86.42	81.26	0.81	0.17	2.89	4.69	0.88	0.11	2.57	4.60
Russian	Taiga	176.63	65	77.75	69.63	0.80	0.13	2.28	5.61	0.89	0.08	2.61	5.45
Scottish Gaelic	ARCOSG	63.70	51	86.57	80.83	0.82	0.16	2.62	4.65	0.87	0.14	2.66	4.36
Serbian	SET	74.26	33	89.18	85.07	0.84	0.17	3.07	4.74	0.89	0.11	2.72	4.68
Slovak	SNK	80.63	42	85.90	79.84	0.88	0.14	2.48	4.58	0.89	0.11	2.76	4.61
Slovenian	SSJ	112.53	22	90.46	87.66	0.87	0.14	3.09	5.01	0.93	0.08	2.83	4.68
Spanish	AnCora	452.74	5	91.90	88.90	0.88	0.07	3.19	4.51	0.94	0.03	2.73	4.12
Spanish	GSD	382.44	10	90.75	87.72	0.87	0.06	3.04	4.33	0.94	0.03	2.58	4.21
Swedish	LinES	55.45	56	85.07	79.98	0.79	0.19	2.72	4.96	0.87	0.13	2.82	4.85
Swedish	Talbanken	66.65	57	83.48	78.97	0.81	0.19	2.67	5.16	0.89	0.13	2.88	5.24
Telugu	MTG	5.08	72	86.47	66.87	0.74	0.16	0.44	4.56	0.60	0.13	0.68	4.40
Turkish	Atis	36.20	27	90.85	88.44	0.90	0.13	1.13	3.48	0.93	0.11	1.59	3.29
Turkish	BOUN	98.21	83	76.09	67.92	0.70	0.16	1.71	5.80	0.81	0.12	2.24	5.76
Turkish	FrameNet	16.33	63	86.49	71.15	0.79	0.13	0.71	3.91	0.65	0.15	1.16	3.94
Turkish	IMST	37.78	88	67.65	58.38	0.58	0.19	1.29	5.69	0.70	0.16	1.83	5.86
Turkish	Kenet	143.29	61	87.14	73.74	0.81	0.13	1.53	4.75	0.79	0.10	1.91	4.69
Turkish	Penn	166.51	50	88.11	77.55	0.83	0.09	1.61	4.50	0.82	0.08	1.89	4.56
Turkish	Tourism	71.24	34	93.57	88.20	0.91	0.09	1.06	5.02	0.89	0.08	1.34	4.89
Ukrainian	IU	92.40	58	85.78	80.97	0.78	0.18	2.88	5.46	0.87	0.11	2.81	5.33
Urdu	UDTB	108.69	43	86.51	80.29	0.83	0.13	2.75	4.32	0.87	0.09	1.88	4.27
Uyghur	UDT	19.26	85	75.25	60.47	0.66	0.20	1.35	4.92	0.66	0.17	1.46	4.96
Vietnamese	VTB	20.29	87	67.44	57.77	0.59	0.24	1.42	5.21	0.71	0.22	1.84	5.14
Western Armenian	ArmTDP	72.09	64	82.99	75.46	0.78	0.19	2.77	5.22	0.82	0.14	2.90	5.24
Wolof	WTB	23.56	70	79.00	71.16	0.72	0.24	2.66	4.37	0.80	0.20	2.61	4.62

Table 1: Results across all treebanks and metrics. Size is measured per thousand tokens. Green cells indicate that the treebank is Top 3 in that category. Red cells correspond to Bottom 3.