Some Languages Seem Easier to Parse Because Their Treebanks Leak

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

Abstract

Cross-language differences in dependency parsing performance are mostly attributed to treebank size, average sentence length, average dependency length, morphological complexity, and domain differences. In this paper I point to a factor not previously discussed: If we abstract away from words and dependency labels, how many graphs in the test data were seen in the training data? I discuss how to compute graph isomorphisms, and show that, treebank size aside, overlap between training and test graphs explains more of the observed variation than standard explanations such as the above.

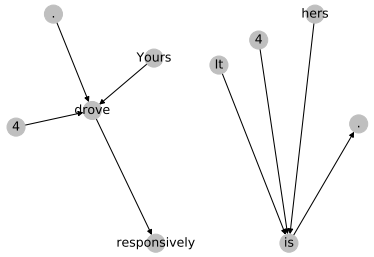

Figure 1: Isomorphic examples from UD-English-Pronouns. Left: *Yours drove responsibly.* Right: *It is hers.* The two sentences are associated with the same unlabeled directed graphs.

1 Introduction

The state of the art in dependency parsing varies a lot across languages: on Polish, the best system in the CoNLL 2018 shared task achieved a labeled attachment score of 94.9% on held-out data; on Basque, the same number was 19.5%. Just a few years ago, a major source of variation was the complexity of the annotation schemes used in the different treebanks; with the Universal Dependencies project,[1] treebanks now follow the same annotation guidelines, but nevertheless, these performance differences persist.[2]

Differences are typically attributed to training set size (Vania et al., 2019), linguistic variation (Nivre et al., 2007), sentence length or average gold dependency length (in the test data) (McDonald and Nivre, 2011), and domain differences between training and test data (Foster et al., 2011). Training set size *is* undoubtedly a very strong predictor of parsing performance, but in this paper, overlap between unlabeled graphs in the training and test sections of a treebank is shown to be more predictive than any of the other factors. Specifically, we compute equivalence classes over unlabeled dependency graphs – directed or undirected – and compute the ratio of trees in the treebanks' test sections that are isomorphic to graphs observed in the training section, i.e., the graph-level train-test leakage, and correlate this number with state-of-the-art performance numbers across languages. To the best of our knowledge, no one has previously considered this predictor of parsing performance, and we show that it is more predictive than factors previously discussed in the literature.

Contribution We present a way to quantify graph-level train-test leakage and an empirical evaluation of it across parsing results for 45 languages;

[1] https://universaldependencies.org/

[2] While Universal Dependencices have made the available dependency treebanks more compatible, treebanks were of course developed using very different protocols; some are automatically or semi-automatically converted from other formalisms, others written with the Universal Dependencies guidelines in mind; some, again, were developed by big teams, some by a single person. While protocol is hard to isolate and study – and while protocol may correlate both positively or negatively with parsing performance, i.e., it is easy to imagine a poorly designed treebank that is easy to parse – the protocol likely has a significant downstream effect on performance; which means we can only hope to explain some of the variance in the experiments below.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 2765–2770,
November 16–20, 2020.

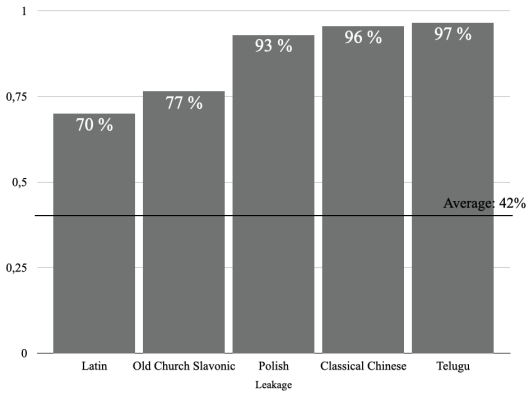

Figure 2: LEAKAGE. The 5 UD treebanks with the most UUG-level train-test leakage.

we show that next to treebank size, graph-level train-test leakage is a better predictor of parsing performance than any of the factors previously considered.

2 Unlabeled Graph Isomorphisms

Exact graph isomorphism is in NP, but it remains an open problem whether it is NP-complete or in P. We use the VF2 algorithm in (Cordella et al., 2001), which is known to be fast in practice with low memory requirements (Foggia et al., 2001). The algorithm proceeds by iteratively expanding a subgraph isomorphism, until this procedure fails, or until the subgraph isomorphism covers the input graphs. We compute isomorphisms over dependency trees in the training set by first reducing the trees to a more abstract graph. In our experiments below, we consider two such reductions: to **undirected, unlabeled graphs** (UUGs; removing labels and edge directions) and to **directed, unlabeled graphs** (DUGs; removing only labels). Once we have computed the isomorphisms, we count how many of the dependency trees in the test data are members of one of these equivalence classes. We then report the fraction of test dependency trees that are isomorphic to at least one dependency tree in the training data. This number can be seen as a metric of graph-level train-test leakage. See Figure 2 for the top 5 most leaking treebanks in the Universal Dependencies project (Version 2.5); the worst has only 3/100 unseen test graphs. In the Appendix, we report the full set of results with UUGs; both for exact computation of isomorphims with VF2, as well as for a heuristic simply matching a set of edge degrees.

3 Usual Suspects

We briefly discuss other factors assumed to be predictive of the performance of dependency parsers.

Treebank size It is trivially true that parser performance depends on treebank size, and it is unsurprising that the correlation is strong. Obviously, if the treebank does not contain *any* training data, supervised parsers will have to resort to blind guessing, and the more data they see, the less variance they have to resolve. That said, it is well established that increasing the size of a treebank often comes with diminishing returns (Sagae et al., 2008). Since treebank size is nevertheless trivially related to parsing performance, we correlate all other factors ϕ *in combination with treebank size* (see §4):

Morphology Previous work has pointed to morphology as a source of lower parsing performance (Tsarfaty et al., 2013; Coltekin and Rama, 2018). In languages with rich morphology, many relations which are expressed implicitly by word order and adjacency in languages like English, are encoded in morphological affixes, which requires subword-level processing to detect (in the tail). Expressing functional information morphologically also allows for a high degree of word-order variation. In our experiments, we use the most predictive morphological feature in WALS[3] and impute the missing values.

Sentence length Parser performance unsurprisingly also depends on input length, i.e., the search space of possible parses (McDonald and Nivre, 2011). This, for example, is why unsupervised dependency parsing has successfully relied on baby steps training (Spitkovsky et al., 2010). We correlate state-of-the-art parser performance with training set size and average test sentence length.

Graph properties McDonald and Nivre (2011) discuss graph properties that seem to correlate with parsing performance. We include average dependency length in our experiments below, which we compute by simply dividing the total length of dependencies by word tokens in the test section.

Open class ratio Nivre and Fang (2017) argue that open word classes (especially nouns and verbs) tend to be harder to attach than other parts of speech, and that languages with many of them will therefore be harder to parse. We therefore evaluate

[3] https://wals.info

	Factors	Explained Variance	Mean Error
	Treebank Size	0.014	0.082
TWO FEATURES	+POS Bigram Perplexity	0.000	0.085
	+Morphology (WALS 21B)	0.000	0.082
	+Open Class Ratio	0.000	0.081
	+$\mathcal{A}$-Distance	0.036	0.078
	+Dependency Length	0.052	0.079
	+Sentence Length	0.170	0.073
	+UUG-ISO	0.222	0.072
	+DUG-ISO	**0.228**	**0.071**

Table 1: EMPIRICAL COMPARISON OF FACTORS. We report the three-fold cross-validation explained variance and mean absolute error of a linear regression model with two features, as well as the baseline of just using a linear regression with treebank size as our only feature.

whether the ratio of nouns and verbs over the total number of tokens in a sentence is predictive of parser performance.

POS bigram perplexity Others have proposed to use the perplexity of a POS bigram language model trained on the treebank's training section and applied to its test section, to predict parser performance (Coltekin and Rama, 2018; Berdicevskis et al., 2018).

Domain divergence Gildea (2001) explore the effect of domain shifts on parsing performance and show that such shifts are often detrimental to the quality of parses. This issue has, since then, been explored in great detail in the domain adaptation literature, but here we simply note that treebanks with train-test divergences may appear harder to parse. In order to compute the impact of train-test divergence on state-of-the-art parsing results, we need to be able to compute it. Several proposals exist in the literature, including Jensen-Shannon divergence (Wu and Huang, 2016), Renyi divergence (Van Asch and Daelemans, 2010), and Wasserstein distance (Shen et al., 2018). We choose to rely on $\mathcal{A}$-distance (Kifer et al., 2004), since it is arguably the most popular divergence measure in domain adaptation, and since we can approximate it efficiently by the accuracy of a linear perceptron trained to discriminate between examples from the train and test splits. Note that Van Asch and Daelemans (2010) explicitly proposed quantifying domain divergence as a way of predicting performance, noting a linear correlation between the two.

4 Empirical Comparison of Factors

We correlate the factors ϕ assumed to influence syntactic dependency parser performance with state-of-the-art performance figures from the CoNLL 2018 shared task, i.e,. the performance of the best performing system per language.[4] See the Appendix for the full statistics. While computing their Pearson's ρ coefficients is standard methodology for validating performance metrics (Lin, 2004; Miculicich Werlen and Popescu-Belis, 2017) and has also been used to evaluate factors predicting system performance (Martin and Foltz, 2004; Søgaard and Haulrich, 2010), this is inadequate in our case: Many factors are potentially covariate, and we are, for example, not interested in factors that correlate strongly with treebank size, e.g., out-of-vocabulary rate or type-token ratio (Kettunen, 2014). Instead we compute the explained variance and mean absolute error of a linear regression model with treebank size and ϕ as input, i.e., $at_s + b\phi + c$ with t_s treebank size and a, b, c learned parameters. We report explained variance and mean absolute error from three-fold cross-validation experiments to avoid overfitting. We make our code publicly available.[5]

Results Our main results are presented in Table 1. Treebank size correlates strongly with parser performance; see the plot in Figure 3 (**Left**). Both **morphological complexity** and **open class ratio** are not very predictive. None of them correlate very strongly with parser performance, and in com-

[4] https://universaldependencies.org/conll18/results-las.html

[5] https://github.com/coastalcph/treebank-leakage

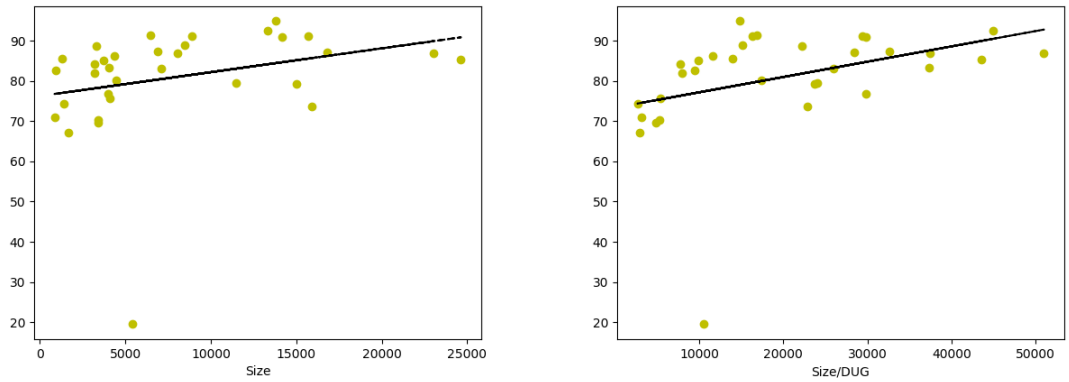

Figure 3: Correlations: Treebank Size (**Left**) and DUG-ISO and Size (**Right**). The outlier in both cases is the Basque treebank.

bination with treebank size, they do not add much predictiver power, it seems. $\mathcal{A}$-distance correlates strongly with parsing performance; the explained variance improves a little, and the error decreases a bit. Average **dependency length** is only weakly, negatively correlated with parsing performance ($\rho \sim 0.067$), a result that is not significant; and the absolute error of the linear regression model decreases only a little from adding the feature; the explained variance improves to 0.05. **Sentence length**, perhaps unsurprisingly, correlates more strongly with parsing performance; and the explained variance of our linear regression model increases a lot from adding this feature.

Graph-level train-test leakage, however, is more predictive of parsing performance than any of these factors. See the correlation of treebank size over DUG-level train-test leakage in Figure 3 (**Right**). It also leads to much better performance of our linear regression model; both in terms of explained variance and mean absolute error. We note that using DUGs to compute the isomorphisms is slightly more predictive than relying on undirected graphs.

5 Related work

The factors evaluated in the above, from Nivre et al. (2007); Van Asch and Daelemans (2010); McDonald and Nivre (2011); Nivre and Fang (2017); Coltekin and Rama (2018); Berdicevskis et al. (2018), were already discussed. A few other factors have been pointed at in the literature that were not applicable to our experiments: Søgaard and Haulrich (2010) show that the perplexity of the derivation orders of a transition-based dependency parser, is also predictive of parser performance. They report Pearson's ρ scores that are considerably higher than those we found. Their study suffers from two biases, though; one imposed by the transition-based parser and the other imposed by the language model used to calculate the perplexity. Moreover, the results they report, are for only the non-convertedd dependency treebanks in the CoNLL 2006 (Buchholz and Marsi, 2006) and CoNLL 2007 (Nivre et al., 2007) treebank releases. These treebanks form a very small set, providing limited statistical support, and, moreover, rely on very different linguistic formalisms and annotation guidelines, leading to very different levels of complexity of derivation. In other words, a comparison would be inconclusive because of the free parameters imposed by the language model and the transition oracle, and the fact that no code is publicly available. Also, their high correlation scores are unlikely to transfer to Universal Dependencies.

6 Discussion and Conclusion

This paper suggested a factor contributing to variance in (universal) dependency parser performance across languages: graph-level train-test leakage in treebanks. This form of leakage can be quantified by computing graph isomorphisms from training sections and counting the ratio of trees in the test sections that are *not* isomorphic with any tree in the training data. I compared this factor to previous attempts to explain variance in parser performance across languages through a series of correlation and linear regression experiments; and showed that graph-level train-test leakage, treebank size aside, is the most predictive factor among those proposed, yet complementary. The result is perhaps

not too surprising, since graph isomorphisms correlate with syntactic constructions, which in turn correlate with the occurrence of linguistic markers and tail linguistic phenomena.[6]

The observation that treebanks leak, quite dramatically, at the graph level, is not only interesting for explaining variance in parser performance. It also suggests a new and improved evaluation methodology: Since language is Zipfian, not only at the level of words, but at the level of phrases (Ha et al., 2002; Williams et al., 2015), standard evaluation methodology relying on random samples (Gorman and Bedrick, 2019; Dodge et al., 2019) is biased toward frequent phenomena. Evaluating only on non-isomorphic trees, i.e., leaving out graphs that have been seen at training time from the test sections of treebanks, would reduce this bias. We hope this is a factor that designers of future syntactic treebanks will take into account. It is an open question whether graph-level train-test leakage is predictive of performance in other sentence-level NLP tasks, i.e., whether the ratio of test sentences whose (predicted) syntactic dependency structure is identical to that of one of our training examples, correlates with state-of-the-art performance.

References

Aleksandrs Berdicevskis, Çağrı Çöltekin, Katharina Ehret, Kilu von Prince, Daniel Ross, Bill Thompson, Chunxiao Yan, Vera Demberg, Gary Lupyan, Taraka Rama, and Christian Bentz. 2018. Using universal dependencies in cross-linguistic complexity research. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 8–17, Brussels, Belgium. Association for Computational Linguistics.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City. Association for Computational Linguistics.

Cagri Coltekin and Taraka Rama. 2018. Exploiting universal dependencies treebanks for measuring morphosyntactic complexity. In *MLC*.

L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. 2001. An improved algorithm for matching large graphs. In *In: 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition, Cuen*, pages 149–159.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. 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 2185–2194, Hong Kong, China. Association for Computational Linguistics.

P. Foggia, C. Sansone, and M. Vento. 2001. A performance comparison of five algorithms for graph isomorphism. In *Proceedings of the 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition*, pages 188–199.

Jennifer Foster, Özlem Çetinoğlu, Joachim Wagner, Joseph Le Roux, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011. From news to comment: Resources and benchmarks for parsing the language of web 2.0. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 893–901, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

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

Kyle Gorman 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, Florence, Italy. Association for Computational Linguistics.

Le Quan Ha, E. I. Sicilia-Garcia, Ji Ming, and F. J. Smit. 2002. Extension of zipf's law to words and phrases. In *ANLP*.

K. Kettunen. 2014. Can type-token ratio be used to show morphological complexity of languages? *Journal of Quantitative Linguistics*, 21(3):223–245.

Daniel Kifer, Shai Ben-David, and Johannes Gehrke. 2004. Detecting change in data streams. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, page 180–191. VLDB Endowment.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Melanie J. Martin and Peter W. Foltz. 2004. Automated team discourse annotation and performance prediction using LSA. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 97–100, Boston, Massachusetts, USA. Association for Computational Linguistics.

[6] If a parser has seen a graph before, it has seen representations similar to those observed at test time. The higher the leakage, the less a parser needs to generalize. Inspecting equivalence classes, members tend to be structurally very similar, and while graph-based parsers, for example, tend to edge-factorize, this still means that context representations are very similar to previously seen ones.

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

Lesly Miculicich Werlen and Andrei Popescu-Belis. 2017. Validation of an automatic metric for the accuracy of pronoun translation (APT). In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 17–25, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 915–932, Prague, Czech Republic. Association for Computational Linguistics.

Kenji Sagae, Yusuke Miyao, Rune Saetre, and Jun'ichi Tsujii. 2008. Evaluating the effects of treebank size in a practical application for parsing. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 14–20, Columbus, Ohio. Association for Computational Linguistics.

Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. 2018. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4058–4065. AAAI Press.

Anders Østerskov Søgaard and Martin Haulrich. 2010. On the derivation perplexity of treebanks. In *Proceedings of Treebanks and Linguistic Theories 9*, NEALT Proceedings Series. ISSN: 1736-6305; null ; Conference date: 03-12-2010 Through 04-12-2010.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2010. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 751–759, Los Angeles, California. Association for Computational Linguistics.

Reut Tsarfaty, Djamé Seddah, Sandra Kübler, and Joakim Nivre. 2013. Parsing morphologically rich languages: Introduction to the special issue. *Computational Linguistics*, 39(1):15–22.

Vincent Van Asch and Walter Daelemans. 2010. Using domain similarity for performance estimation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 31–36, Uppsala, Sweden. Association for Computational Linguistics.

Clara Vania, Yova Kementchedjhieva, Anders Søgaard, and Adam Lopez. 2019. A systematic comparison of methods for low-resource dependency parsing on genuinely low-resource languages. 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 1105–1116, Hong Kong, China. Association for Computational Linguistics.

Jake Ryland Williams, Paul R. Lessard, Suma Desu, Eric M. Clark, James P. Bagrow, Christopher M. Danforth, and Peter Sheridan Dodds. 2015. Zipf's law holds for phrases, not words. In *ANLP*.

Fangzhao Wu and Yongfeng Huang. 2016. Sentiment domain adaptation with multiple sources. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 301–310, Berlin, Germany. Association for Computational Linguistics.