Himanshu Yadav*, Samar Husain and Richard Futrell

Do dependency lengths explain constraints on crossing dependencies?

https://doi.org/10.1515/lingvan-2019-0070
Received October 15, 2019; accepted October 23, 2020

Abstract: In syntactic dependency trees, when arcs are drawn from syntactic heads to dependents, they rarely cross. Constraints on these crossing dependencies are critical for determining the syntactic properties of human language, because they define the position of natural language in formal language hierarchies. We study whether the apparent constraints on crossing syntactic dependencies in natural language might be explained by constraints on dependency lengths (the linear distance between heads and dependents). We compare real dependency trees from treebanks of 52 languages against baselines of random trees which are matched with the real trees in terms of their dependency lengths. We find that these baseline trees have many more crossing dependencies than real trees, indicating that a constraint on dependency lengths alone cannot explain the empirical rarity of crossing dependencies. However, we find evidence that a combined constraint on dependency length and the rate of crossing dependencies might be able to explain two of the most-studied formal restrictions on dependency trees: gap degree and well-nestedness.

Keywords: crossing dependencies; dependency length; dependency treebanks; efficiency; language processing; syntax

1 Introduction

Two goals of linguistics are to characterize natural languages as formal systems, and also as codes for communication. The **efficiency hypothesis**, pursued by linguists for over a century, claims that these two goals are related, and that the formal properties of natural language are best explained in terms of maximizing the amount of information transferred while minimizing the complexity of language production and comprehension (Chomsky 2005; Ferrer-i-Cancho and Solé 2003; Gibson et al. 2019; Haspelmath 2008; Hawkins 1994; Hockett 1960, 2004, 2014; von der Gabelentz 1901; Zipf 1949). In recent years, it has become possible to test such theories quantitatively using corpora of many languages.

Within the framework of the efficiency hypothesis, one influential proposal is **dependency length minimization** (DLM). DLM is the idea that words in syntactic dependencies are under a pressure to be close to each other in linear order. Syntactic dependencies are relations between words as illustrated in Figure 1. The connection between DLM and efficiency is that when dependency lengths are minimized, more memory-efficient generation and parsing is possible (Gibson 1998). For recent reviews on DLM, see Dyer (2017), Liu et al. (2017), and Temperley and Gildea (2018). DLM has demonstrated considerable power for explaining a number of language universals involving word order, such as Greenberg's harmonic word order correlations (Greenberg 1963; Hawkins 1994), as well as exceptions to them (Gildea and Temperley 2010; Temperley 2008).

In this work, we examine the claim that DLM might be the underlying factor behind an even more basic property of syntax: the distribution of **crossing dependencies**. When dependency arcs are drawn above a sentence as in Figure 1, they may cross. But empirically, in real trees, this happens rarely and with formal restrictions (Ferrer-i-Cancho et al. 2018; Havelka 2007; Nivre and Nilsson 2005). As we review below, the

***Corresponding author: Himanshu Yadav**, University of Potsdam, Potsdam, Germany, E-mail: hyadav@uni-potsdam.de
Samar Husain, Indian Institute of Technology Delhi, Delhi, India
Richard Futrell, University of California Irvine, Irvine, USA

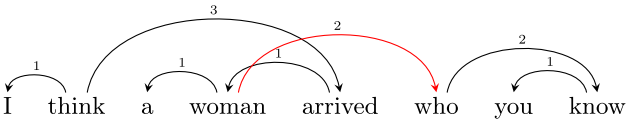


Figure 1: An example dependency tree. Arrows point from heads to dependents. Each dependency arc is labeled with its **dependency length**: the distance from the head to the dependent, measured in words. This tree has one crossing dependency. All crossing dependencies in all figures are marked in red. It has gap degree 1, edge degree 1, end-point crossings 1, HDD 2, and is well-nested.

properties of these crossing dependencies are of significance for both formal linguistics and natural language processing.

Recently, it has been proposed that the rarity of crossing dependencies is a side-effect of DLM, such that there is no need to posit additional constraints against crossing dependencies in order to explain the distribution of crossing arcs observable in real linguistic trees (Ferrer-i-Cancho 2006, 2016). Here we examine whether the distribution of crossing dependencies observable in dependency treebanks can be explained in terms of constraints on dependency lengths, or whether other factors are required.

Results show that constraints on dependency lengths do not suffice to explain the rarity of crossing dependencies observed in corpora. However, we show that a combined constraint on both dependency length and the rate of crossing dependencies might be able to explain two of the most-discussed formal restrictions on crossing dependencies: gap degree and well-nestedness.

2 Background

The current work brings together the fields of formal language theory, natural language processing, graph theory, and corpus linguistics. Here we review previous work on crossing dependencies from these different perspectives. We show the importance of crossing dependencies for the formal characterization of natural language and for the development of natural language processing algorithms, and review previous results about the relationship between dependency length and crossing dependencies, both from theoretical and empirical perspectives.

2.1 Crossing dependencies and formal grammar

A striking result from the last three decades of research in formal linguistics is that the formal characterization of the syntax of human language (in the sense of Chomsky 1959; Chomsky and Schützenberger 1963; Hopcroft and Ullman 1979) is deeply related to the distribution of crossing dependencies in dependency trees.

In syntactic theory, crossing dependencies correspond to **displacement** phenomena, and are modeled using a distinct kind of structure from non-crossing dependencies. For example, in phrase-structure frameworks, displacement phenomena are modelled using slash-passing (Pollard and Sag 1994; Steedman and Baldridge 2011); in Minimalist frameworks, they are modeled using a distinct structure-building operation MOVE (Boston et al. 2010; Chomsky 1995; Michaelis 1998).

The formal mechanisms which have been invoked to describe crossing dependencies turn out to be crucial for the formal language-theoretic characterization of natural language, because they involve operations that go beyond context-free grammar. To see the connection between crossing dependencies and formal language theory, first note that dependency trees with no crossing dependencies—which are called **projective**—correspond exactly to structures generated by lexicalized context-free phrase-structure grammars (Marcus 1965). So if there were no crossing dependencies in linguistic trees, then natural language would be context-free. Crossing dependencies indicate exactly those cases where natural language moves beyond

context-freeness, and limitations on crossing dependencies indicate exactly how and when natural language can deviate from context-freeness.

Human language is known to be non-context-free (Shieber 1985), but it does not appear to be formally unrestricted. The consensus is that human language occupies a language class called **mildly context-sensitive** between the context-free and fully context-sensitive languages (Joshi et al. 1991; Weir 1988). These mildly context-sensitive languages are defined by formal constraints which turn out to be equivalent to constraints on crossing dependencies. In particular, most mildly context-sensitive formalisms have bounds on a quantity called **gap degree**, which has been shown to relate to formal restrictions on crossing dependencies by Kuhlmann and Nivre (2006). Some mildly context-sensitive formalisms also induce a constraint called **well-nestedness**, which can also be reduced to constraints on crossing dependencies (Bodirsky et al. 2005). For linguistic considerations involving these constraints, see Maier and Lichte (2009), Chen-Main and Joshi (2010), Mambrini and Passarotti (2013), Miletic and Urieli (2017), and Yadav et al. (2017).

Above, we sketched the connection between crossing dependencies and formal syntax. Because of this connection, it is particularly interesting that the distribution of crossing dependencies in natural language might be explained by DLM. Any theory which is capable of explaining constraints on crossing dependencies is capable of explaining and characterizing the formal language class of natural language, thus answering some of the most basic questions in linguistics.

2.2 Crossing dependencies and parsing algorithms

The nature of formal grammars is intimately connected to parsing algorithms, both in the context of psycholinguistics and computational linguistics. For example, based on behavioral experiments investigating human processing difficulty in crossing dependencies and embedded dependencies for Dutch and German native speakers, Bach et al. (1986) argued that a push-down automaton cannot form the basis for natural language parsing by humans cross-linguistically. Subsequently, Joshi (1990) proposed an embedded push-down automaton (EPDA) to account for the results of Bach et al. (1986) and made the connection between EPDA and formal grammars such as Tree-Adjoining Grammars, Categorial Grammars and Head Grammars.

Given this context, crossing dependencies are also of interest in the design of dependency parsers in natural language processing. Efficient parsing algorithms are only possible when there are formal constraints on crossing dependencies (Eisner and Giorgio 1999; Gómez-Rodríguez et al. 2010; Kuhlmann 2013; Pitler et al. 2013).

2.3 Crossing dependencies and dependency length minimization

Ferrer-i-Cancho (2006) proposed that the apparent rarity of crossing dependencies is a consequence of DLM. This proposal is based the following graph-theoretic observation: when one arranges the nodes of a dependency tree in order to minimize dependency length—i.e., solving the Optimal Linear Arrangement (OLA) problem from graph theory (Chung 1984; Gildea and Temperley 2007; Harper 1964; Hochberg and Stallmann 2003; Park and Levy 2009; Shiloach 1979)—one gets trees with nearly zero crossing dependencies for sentences of reasonable length. In subsequent work, Ferrer-i-Cancho (2014, 2016) has given analytical formulas to approximately predict the number of crossing dependencies in arbitrary dependency trees given knowledge of dependency lengths, finding that the formulas gave only slight overestimations of the number of crossing arcs in selected linguistic trees. Relatedly, Ferrer-i-Cancho and Gómez-Rodríguez (2016) have documented that the rate of crossing dependencies in dependency treebanks cannot be explained by a specific bound on the number of crossing arcs allowed per sentence length, and is instead a function of the lengths of dependencies in the tree.

Making explicit the connection with formal language theory, Gómez-Rodríguez et al. (2019) have examined dependency length in random linear arrangements (RLAs) of trees from dependency treebanks, while

controlling for proposed formal restrictions on crossing dependencies. They find that formal constraints on crossing dependencies tend to reduce the dependency lengths in such trees, arguing that DLM is the factor that explains the apparent formal language class for natural language. As we will see, we take the opposite approach: we control for the empirical distribution of dependency length and look at the formal properties of the resulting RLAs. This enables us to determine whether dependency length as an independent variable is sufficient to explain formal properties as a dependent variable.

In work closely related to ours, Lu and Liu (2016) have presented some complicating evidence for the hypothesis that DLM can explain the rarity of crossing dependencies. They find that minimizing mean dependency length in random trees does correlate with a reduction in crossing dependencies, but that realistic rates of crossing dependencies are only attained for very small values of mean dependency length, much smaller than what is found in natural languages. In the current work, we present conclusive evidence that DLM (alone) cannot explain the actual rate of crossing dependencies found in natural language.

3 Methods

We are interested in explaining two properties of dependency trees: (1) the rate of crossing dependencies (average number of crossing arcs at each sentence length), and (2) the formal properties of crossing dependencies. To do this, we will compare the crossing rate and formal crossing properties in real trees drawn from dependency treebanks against a baseline of random trees which are generated under explicit constraints on dependency length.

3.1 Definitions

A **dependency tree** t is a directed labeled tree with nodes corresponding to words and arcs $\langle h, d\rangle$ called **dependencies**, where h identifies a **head** node and d identifies a **dependent** node. We say a dependency $\langle h, d\rangle$ in tree t is **crossing** if there exists a node m intervening between h and d in the linear order of t such that m is not a transitive descendant of h. So the dependency $\langle$*woman, who*$\rangle$ in Figure 1 is crossing, because *arrived* is not a descendant of *woman*. On the other hand, the dependency $\langle$*think, arrived*$\rangle$ is not crossing, because all the intervening nodes are descendants of *think*.

A variety of formal constraints on crossing dependencies have been proposed in the formal grammar and dependency parsing literature. Together, we call these constraints **formal crossing constraints**. We study the following formal crossing constraints, with formal definitions given in Supplementary Materials Section S2: *gap degree* (Figure 2), *well-nestedness* (Figure 3), *edge degree* (Figure 4), *end-point crossings* (Figure 4), and *heads' depth difference* (Figure 5). Gap degree, well-nestedness, and end-point crossings are relevant for the efficiency of exact parsing algorithms. Edge degree and heads' depth difference have also been implicated in human parsing difficulty (Yadav et al. 2017, 2020).

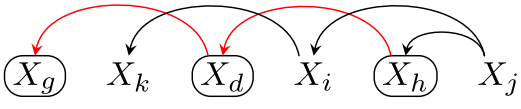


Figure 2: Example of a configuration with gap degree 2.

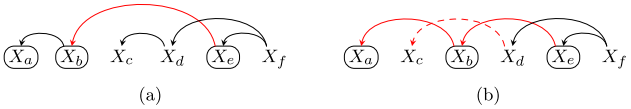


Figure 3: Example of well-nested and ill-nested tree configurations.
(a) Well-nested. (b) Ill-nested.

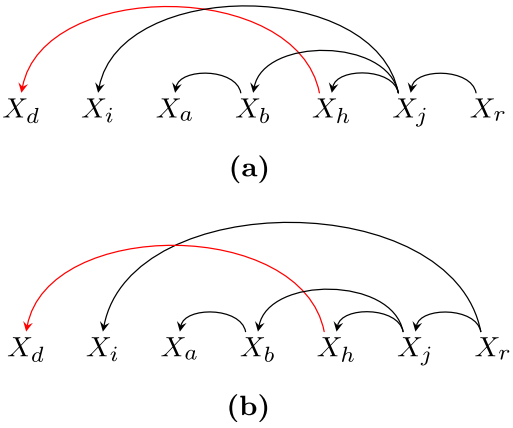


Figure 4: Examples of trees with edge degree 2 and end-point crossings 1 and 2.
(a) Edge degree 2, end-point crossings 1. **(b)** Edge degree 2, end-point crossings 1.

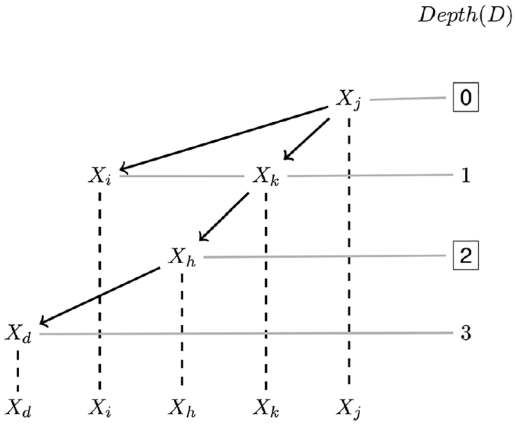


Figure 5: Example of tree configuration with HDD 2.

3.2 Baselines

3.2.1 Random trees with controlled dependency length

Our first baseline is called **random trees**. This baseline consists of samples from a uniform distribution over tree structures of a given length, under the constraint that each random tree must have the same distribution of dependency lengths as some attested dependency tree.

In order to generate random trees with controlled dependency length, we first define the **dependency length (DL) sequence** of a dependency tree as the list of lengths of each dependency, sorted in order of increasing length. For example, the DL sequence for the tree in Figure 1 is [1, 1, 1, 1, 2, 2, 3].[1] To generate the random trees baseline, we take each attested tree t with n words from a dependency treebank, and generate a random tree of length n with the same DL sequence as t. For more details, see Supplementary Materials Section S2. For example, a random tree with the same DL sequence as in Figure 1 is shown in Figure 6.

Following previous work (Yadav et al. 2019), we generate these controlled random trees by rejection sampling. Given an attested dependency tree t with length n and DL sequence d, we repeatedly generate random trees of length n until we find one matching the desired DL sequence d. This procedure is slow because very few random trees match the desired DL sequence, especially for long sentences. Therefore we are

1 Dependency trees as found in dependency treebanks usually have a virtual "root node" at the beginning of each sentence. We do not consider these root nodes when calculating DL sequence, but we do consider them when calculating the number of crossing arcs and their properties.

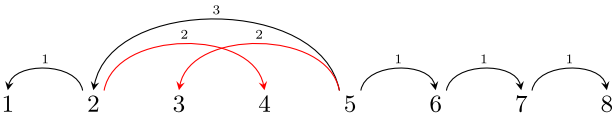


Figure 6: A random tree matched with Figure 1 for length and DL sequence [1, 1, 1, 1, 2, 2, 3] (see text). This tree has two crossing dependencies (marked in red). It has gap degree 1, edge degree 1, end-point crossings 1, HDD 1, and is ill-nested.

currently restricted to examining sentences of less than 12 words. For some evidence that the patterns we report here are likely to hold qualitatively for longer sentences as well, see Section 4.1.

Our baseline trees control for the complete empirical distribution of dependency lengths. This means that *any* function of the set of dependency lengths will be identical between real trees and our baseline trees.

3.2.2 Random linear arrangements with controlled dependency length

As a second baseline, we also generate **random linear arrangements** (RLAs) of original trees, again controlling the DL sequence. A random linear arrangement of a labeled tree t is a permutation of the nodes of t, which when applied to dependency trees scrambles the order of words while keeping the dependency relationships among the words the same. For example, a random linear arrangement of the tree from Figure 1 is shown in Figure 7.

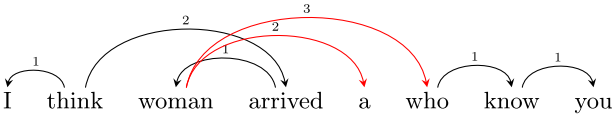


Figure 7: A random linear arrangement of the tree in Figure 1 with the same DL sequence of [1, 1, 1, 1, 2, 2, 3]. This tree has two crossing dependencies (marked in red). This tree has gap degree 1, edge degree 1, end-point crossings 1, HDD 1, and is well-nested.

These random linear arrangements control for all the topological properties of the original trees, such as their depth, arity, hubbiness, etc., which are known to constrain the possible number of crossing arcs (Ferrer-i-Cancho et al. 2018). These constrained random linear arrangements are generated by the same rejection sampling procedure as for random trees.

3.2.3 Controlling crossing rate

We also experiment with baselines that simultaneously control for dependency length and crossing rate. In this case, we are interested in the formal crossing constraints in the resulting random trees. In these baselines, given an attested tree t of length n from a treebank, we randomly generate (again by rejection sampling) a random tree of length n with the same DL sequence as t and the same number of crossing dependencies as t. For results on random trees controlling only the rate of crossing dependencies, but not the dependency length distribution, see Yadav et al. (2019).

3.3 Statistical methods

We are interested in whether real trees differ from baseline trees in their crossing rates and formal crossing constraints. Therefore, our dependent variables are crossing rates and rates of violations of formal crossing constraints. These dependent variables can be seen as functions of sentence length, tree depth, and/or arity: for example, it may be the case that the distribution of gap degree differs between real and random trees as a function of tree depth but not sentence length. When some tree property is not significantly different between real and baseline trees, then we have no evidence that an additional constraint is needed to explain that

property. On the other hand, if the tree property is significantly different, then we have evidence that there must be some constraint, beyond what has already been controlled, to explain that property.

To test whether there is statistical evidence for a difference between real and baseline trees, we use Poisson regression, which models rates of events. Given a collection of trees, some of them real and some of them baseline, we fit a regression to predict formal properties of the tree. For example, to predict gap degree as a function of sentence length, we fit values β to minimize the square of the error ϵ:

$$\log y_i = \beta_0 + \beta_l l_i + \beta_r 1_{\text{real}} + \beta_{rl} l_i 1_{\text{real}} + \epsilon, \quad (1)$$

where y_i is the gap degree of the ith sentence, l_i is the length of the ith sentence, and 1_{real} is an indicator variable with value 1 for a real tree and 0 for a baseline tree. We evaluate whether there is a significant difference between real and random trees by checking whether a regression as in Eq. (1) is a significantly better fit to the data than a regression lacking the terms in 1_{real} according to a χ^2 test on the likelihood ratio. For full details on these regressions, see the Supplementary Materials Sections S3 and S4.

When fitting regressions to corpus data, in order to achieve the best statistical power, we sometimes combine trees from treebanks of multiple languages. When we do so, we add random intercepts by language to our regression to account for inter-language differences (Baayen et al. 2008; Barr et al. 2013).

3.4 Data sources

We use treebanks from Surface-syntactic Universal Dependencies (SUD) v.2.4 (Gerdes et al. 2018, 2019), based on treebanks originally annotated as part of the Universal Dependencies (UD) project (Nivre 2015, 2019) which have been converted to reflect syntactic dependencies rather than the more semantic dependencies favored by UD. We chose to use SUD because theories such as DLM have nearly always been formulated in terms of surface syntactic dependencies (Liu et al. 2017; Temperley and Gildea 2018) (for a discussion of the difference between UD and SUD from the perspective of dependency length, see Yan and Liu 2019).[2] Following previous work using dependency treebanks for linguistic-typological research, we remove all punctuation nodes from dependency trees.

3.5 Languages tested

We test on a total of 52 languages, taking the largest SUD treebank per language and excluding treebanks with less than 500 sentences and ancient languages.[3] We performed all studies initially on 19 languages as a pilot study, then preregistered our methods and predictions before proceeding to test on the remaining languages.[4]

4 Results

4.1 Controlling only dependency length

As shown in Figures 8 and 9, we find that baseline trees with controlled DL sequence have many more crossing dependencies than real trees. The overall rate of crossing dependencies is significantly higher in baseline trees than in real trees ($p < 0.001$ for both random trees and random linear arrangements). This pattern also holds in individual regressions per language ($p < 0.001$ for every language, for both random trees and random linear arrangements).[5]

2 In preliminary work, we also experimented with the original UD treebanks. We found that these have a much lower rate of crossing dependencies than SUD treebanks, by as much as a factor of 10, often due to the handling of auxiliary verbs. The results regarding formal crossing constraints are qualitatively similar between the two annotation styles.

3 We excluded the following ancient languages: Latin, Ancient Greek, Sanskrit, Old Church Slavonic, Old Russian, Old French.

4 Preregistration available at https://aspredicted.org/ii67u.pdf.

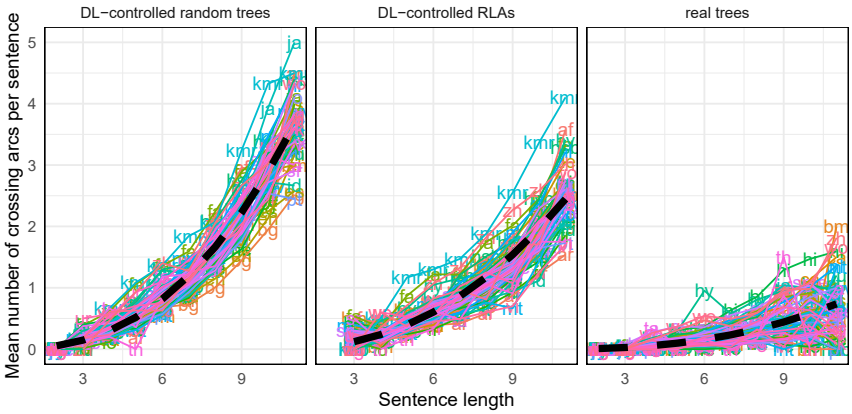


Figure 8: Mean numbers of crossing dependencies per sentence length, for real and baseline trees matched in dependency length. RLA stands for random linear arrangement. Languages are indicated by their ISO codes as given in the SUD corpus files (ISO 639-3 2007). The dashed black lines show the fit of Poisson regressions predicting the number of crossing dependencies from the log sentence length.

Our study is limited in that we can only generate baseline trees for short sentences. However, we believe that the results we have reported here will continue to hold qualitatively for longer sentences as well. As an example, in Figure 10, we show the rates of crossing dependencies for real languages up to sentence length 30, along with the regressions that we fit to sentences of length up to 11. The low growth rate of crossing dependencies in real trees continues for longer sentences. If the trends seen in this plot continue, then the rate of crossing dependencies in the baseline trees will likely continue to be much higher than that in the real trees.

4.2 Controlling dependency length and crossing rate

Now we turn to the results about comparing formal crossing constraints (gap degree, edge degree etc.) in real trees versus baseline trees. The critical p-values are summarized in Table 1; a low p-value indicates a significant difference between real and random trees, and a high p-value indicates weak or no evidence for a difference. The results comparing against random linear arrangements are summarized in Table 2. Full regression results are given in the Supplementary Materials in Table 2.

From these tables, a striking and consistent pattern emerges: well-nestedness is not significantly different between real and baseline trees, and gap degree is only significantly different between real trees and random tree structures. When comparing real trees and random linear arrangements, the gap degree distributions are statistically nearly indistinguishable. On the other hand, the remaining three formal crossing constraints—edge degree, end-point crossings, and HDD—are dramatically different when comparing real trees against any random baseline.

4.3 Interpretation

When we do not control crossing rate, our baselines show what dependency trees would look like if they were constrained only to have a certain distribution of dependency lengths. We find that such trees have many more crossings than real trees, consistently across languages, indicating that the distribution of dependency lengths in natural language does not suffice to explain the low rate of crossing dependencies. In other words, dependency length can be minimized to the point that we actually find in natural language, without reducing crossing arcs to the rate that we actually find in natural language.

5 See Supplementary Materials Sections S3 and S4 for model specifications and detailed results.

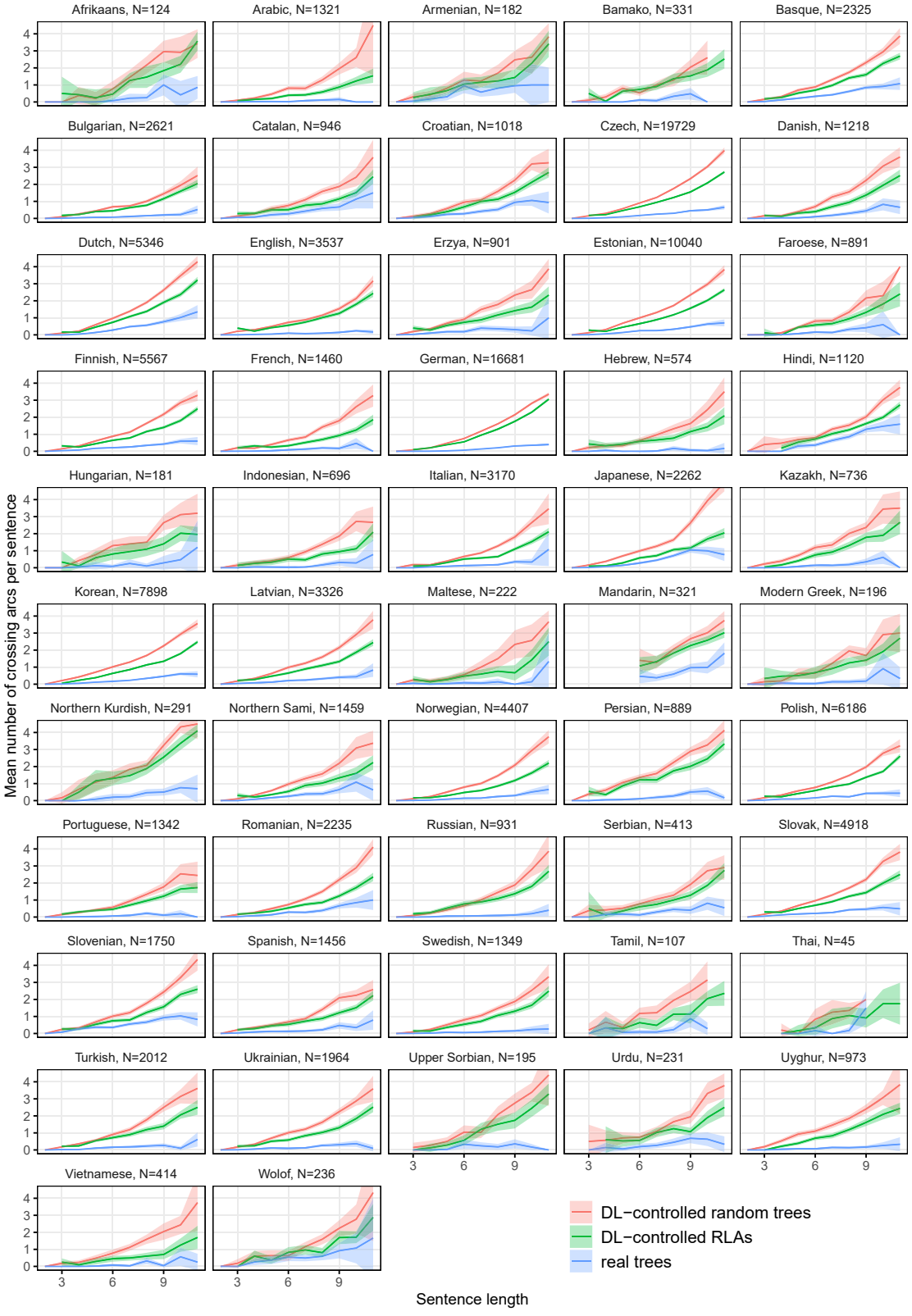


Figure 9: Mean numbers of crossing dependencies per sentence length, per language, for real trees (blue) and baseline trees matched in dependency length (red and green). Shaded areas show 95% confidence intervals of the mean. *N* represents the total number of trees for which we generated baseline trees.

Figure 10: Crossing rates for real sentences up to length 30, along with regression fits from Figure 8 for sentences of length <12 for real and baseline trees.

Table 1: *p*-Values on the difference between real trees and random trees controlled for dependency length and crossing rate. For statistical methods see Section 3.3. Non-significant values in italics.

Crossing constraint	~Length	~Depth	~Arity	~∑DL
Gap degree	<0.001	<0.001	<0.001	<0.001
Well-nestedness	*0.66*	0.03	*0.95*	*0.61*
Edge degree	<0.001	<0.001	<0.001	<0.001
End-point crossings	<0.001	<0.001	<0.001	<0.001
HDD	<0.001	<0.001	<0.001	<0.001

However, when we do control crossing rate in addition to dependency length, we find that the baseline trees have very similar gap degree and well-nestedness to real trees. The result suggests that well-nestedness may not be a true constraint on dependency trees, but rather an epiphenomenon arising from more generic restrictions on dependency length and crossing rate. Gap degree, in turn, may be an epiphenomenon arising from restrictions on dependency length, crossing rate, and tree depth, as indicated by the fact that gap degree is significantly different between real trees and random tree structures, but not significantly different between real trees and random linear arrangements of the real trees.

It might be surprising that DL-controlled baseline trees show a different distribution of crossing dependencies from the real trees. It turns out that the typical distribution of dependency length in real trees—which tends toward short dependencies—is compatible with high edge degree, high end-point crossings, and high HDD, even though these are not found in real trees. For example, in order to have an edge degree of 4 and end-point crossings 3, the minimal DL sequence requirement is [1, 1, 1, 2, 3, 3, 5] (see Figure 11(b)). Such a dependency length distribution is not uncommon in real language trees. To make the example concrete, a

Table 2: *p*-Values on the difference between real trees and random linear arrangements controlled for dependency length and crossing rate. For statistical methods see Section 3.3. Non-significant values in italics.

Crossing constraint	~Length	~Depth	~Arity	~∑DL
Gap degree	*0.72*	*0.66*	*0.52*	*0.13*
Well-nestedness	*0.98*	*0.98*	*0.99*	*0.99*
Edge degree	<0.001	<0.001	<0.001	<0.001
End-point crossings	<0.001	<0.001	<0.001	<0.001
HDD	<0.001	<0.001	<0.001	<0.001

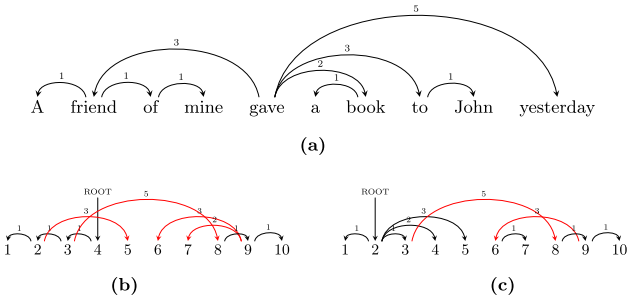


Figure 11: Demonstration of high rate of crossings, high edge degree, and high end-point crossings in random trees that have the same DL sequence as the real tree of a simple English sentence. (a) is a real language tree with DL sequence [1, 1, 1, 1, 1, 2, 3, 3, 5]. (b) and (c) are DL-controlled random trees matched for length and DL sequence [1, 1, 1, 1, 1, 2, 3, 3, 5]. Tree (b) has four crossing dependencies (marked in red); edge degree 4 and endpoint crossings 3 (see arc 3→8). Tree (c) has four crossing dependencies; edge degree 3 and endpoint crossings 2 (see arc 3→8).

simple sentence like *A friend of mine gave a book to John yesterday* has DL sequence [1, 1, 1, 1, 1, 2, 3, 3, 5]; as shown in Figure 11, the corresponding random trees with same dependency length distribution could easily have end-point crossings up to 3 and edge degree up to 4.

4.4 Effect of word order variability

As shown in Figure 9, the rate of crossing dependencies in some languages (e.g., Basque) seems close to that in random baselines, while other languages are dramatically different from the baselines (e.g., English). This suggests that languages could have different distributions of crossing dependencies based on the degree of word order flexibility permitted. Here we explore if word order flexibility in a corpus has any influence on the distribution of crossing dependencies, building on work by Liu (2010), who studies proportions of crossing dependencies in crosslinguistic treebanks from a typological perspective. In particular, we test whether languages with more word order freedom have more crossing dependencies, and whether they have different distributions of the formal crossing constraints.

We operationalize word order freedom as the degree of variability in the order of subject, verb, and object, quantified using the entropy of orders of main verbs along with their *subj* and *comp:obj* dependents, as in Futrell et al. (2015b). We divide corpora into high-flexibility and low-flexibility groups according to whether this entropy is greater than or lower than the average, respectively.[6]

As shown in Table 3, the rate of crossing dependencies with respect to sentence length is significantly different between languages with high word-order freedom and languages with low word-order freedom ($p = 0.01$). Table 3 also shows that the two groups of languages are also distinct configurationally with regard to various crossing constraints.

6 We note that each corpus contains texts from different genres, which may affect the observed word order flexibility. In addition, while the results consider all the dependency relations in the treebank, the current classification is based on verb–argument dependencies. A classification based on additional dependency types will be taken up in future research. *Corpora in the high-flexibility group*: Afrikaans, Arabic, Armenian, Basque, Catalan, Croatian, Czech, Danish, Dutch, Estonian, Galician, German, Hungarian, Italian, Latvian, Polish, Romanian, Slovak, Spanish, Ukrainian, and Urdu. *Corpora in the low-flexibility group*: Bulgarian, Chinese, English, Finnish, French, Greek, Hebrew, Hindi, Indonesian, Irish, Italian, Japanese, Korean, North Sami, Norwegian, Persian, Portuguese, Russian, Serbian, Swedish, Tamil, Turkish, and Vietnamese.

Table 3: *p*-Values on the difference between crossing constraints in high-flexibility versus low-flexibility languages as a function of sentence length, tree depth, tree arity, and sum dependency length. For statistical methods see Section 3.3.

Crossing constraint	~Length	~Depth	~Arity	~ΣDL
Number of crossings	0.01	<0.001	<0.001	<0.001
Gap degree	0.04	<0.001	<0.001	<0.001
Well-nestedness	0.01	0.04	0.007	<0.001
Edge degree	0.02	0.006	<0.001	<0.001
End-point crossings	0.02	0.005	<0.001	<0.001
HDD	0.01	<0.005	<0.001	<0.001

5 Conclusion

We assessed the promising hypothesis that the distribution of crossing dependencies in natural language might be explained by a simple functionally-motivated principle of dependency length minimization (DLM). We found that constraints on dependency lengths alone cannot suffice to explain the low rate of crossing dependencies observable in dependency treebanks. Specifically, we found that there are many possible tree structures and word orders that have the same dependency lengths as real trees and yet have many more crossing dependencies than real trees. Some further constraint on trees is necessary to explain the low rate of crossing dependencies in natural language, although not necessarily a direct constraint against crossing dependencies.

We have found that DLM alone does not fully explain the rate of crossing dependencies. However, we stress that DLM retains strong explanatory power for word order correlations, length-based word order preferences, and the overall distribution of dependency lengths in corpora (Chen and Gerdes 2019; Ferrer-i-Cancho 2004; Futrell et al. 2015a; Liu 2008). So the results here should not be taken as evidence against the DLM hypothesis, although they do imply that some other factor beyond DLM is required to explain the distribution of crossing dependencies in particular.

Our results are also favorable for DLM in the following way: we find that controlling for dependency length *and* the rate of crossing dependencies allows us to potentially explain the distribution of two formal properties of crossing dependencies—gap degree and well-nestedness—at least in short sentences. These two properties have been used to define the mildly context-sensitive hierarchy; therefore, our results suggest that this formal language class might be fully explainable in terms of functionally-motivated constraints such as DLM and another currently-poorly-understood constraint that lowers the rate of crossing dependencies.

Finally, we find evidence that the rate of crossing dependencies varies significantly based on word order flexibility of a language. The results also show that the two groups of languages are different with regard to various crossing constraints. At the same time, the rate of crossings in both the groups remains well below the rate in the random baselines. These results leave open the possibility of differential thresholds of crossing constraints across the two groups. A more detailed investigation on the implication of typological differences for crossing constraints will be taken up in future research.

Acknowledgments: We thank the three anonymous reviewers for helpful suggestions. This work was supported by a gift from the NVIDIA Corporation.

References

Baayen, R. Harald, Douglas J. Davidson & Douglas M. Bates. 2008. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language* 59(4). 390–412.

Bach, Emmon, Colin Brown & William D. Marslen-Wilson. 1986. Cross and nested dependencies in German and Dutch: A psycholinguistic study. *Language and Cognitive Processes* 1(4). 249–262.

Barr, Dale J., Roger P. Levy, Christoph Scheepers & Harry J. Tily. 2013. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language* 68(3). 255–278.

Bodirsky, Manuel, Marco Kuhlmann & Mathias Möhl. 2005. Well-nested drawings as models of syntactic structure. In *Tenth conference on formal grammar and ninth meeting on mathematics of language, Edinburgh*, 195–203.

Boston, Marisa Ferrara, John T. Hale & Marco Kuhlmann. 2010. Dependency structures derived from minimalist grammars. In *Proceedings of the 10th and 11th biennial conference on the mathematics of language*, 1–12. Berlin: Springer-Verlag.

Chen, Xinying & Kim Gerdes. 2019. The relation between dependency distance and frequency. In *Proceedings of the first workshop on quantitative syntax*, 75–82. Paris: Association for Computational Linguistics.

Chen-Main, Joan & Aravind K. Joshi. 2010. Unavoidable ill-nestedness in natural language and the adequacy of tree local-MCTAG induced dependency structures. In *Proceedings of the 10th international conference on tree adjoining grammars and related formalisms (TAG+10)*, 53–60. New Haven: Yale University.

Chomsky, Noam. 1959. On certain formal properties of grammars. *Information and Control* 2(2). 137–167.

Chomsky, Noam. 1995. *The minimalist program*. Cambridge, MA: MIT Press.

Chomsky, Noam. 2005. Three factors in language design. *Linguistic Inquiry* 36(1). 1–61.

Chomsky, Noam & Marcel P. Schützenberger. 1963. The algebraic theory of context free languages. In P. Braffot & D. Hirschberg (eds.), *Computer programming and formal languages*, 118–161. Amsterdam: North Holland.

Chung, Fan-Rong King. 1984. On optimal linear arrangements of trees. *Computers & Mathematics with Applications* 10(1). 43–60.

Dyer, William E. 2017. *Minimizing integration cost: A general theory of constituent order*. Davis, CA: University of California, Davis Dissertation.

Eisner, Jason & Giorgio Satta. 1999. Efficient parsing for bilexical context-free grammars and head automaton grammars. In *Proceedings of the 37th annual meeting of the association for computational linguistics*, 457–464. College Park: Association for Computational Linguistics.

Ferrer-i-Cancho, Ramon. 2004. Euclidean distance between syntactically linked words. *Physical Review E* 70. 056135.

Ferrer-i-Cancho, Ramon. 2006. Why do syntactic links not cross? *Europhysics Letters* 76(6). 1228.

Ferrer-i-Cancho, Ramon. 2014. A stronger null hypothesis for crossing dependencies. *Europhysics Letters* 108(5). 58003.

Ferrer-i-Cancho, Ramon. 2016. Non-crossing dependencies: Least effort, not grammar. In Alexander Mehler, Andy Lücking, Sven Banisch, Philippe Blanchard & Barbara Job (eds.), *Towards a theoretical framework for analyzing complex linguistic networks*, 203–234. Berlin: Springer.

Ferrer-i-Cancho, Ramon & Carlos Gómez-Rodríguez. 2016. Crossings as a side effect of dependency lengths. *Complexity* 21(S2). 320–328.

Ferrer-i-Cancho, Ramon, Carlos Gómez-Rodríguez & Juan Luis Esteban. 2018. Are crossing dependencies really scarce? *Physica A: Statistical Mechanics and its Applications* 493. 311–329.

Ferrer-i-Cancho, Ramon & Ricard V. Solé. 2003. Least effort and the origins of scaling in human language. *Proceedings of the National Academy of Sciences* 100(3). 788.

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

Futrell, Richard, Kyle Mahowald & Edward Gibson. 2015b. Quantifying word order freedom in dependency corpora. In *Proceedings of the third international conference on dependency linguistics (DepLing 2015)*, 91–100. Uppsala: Uppsala University.

Gerdes, Kim, Bruno Guillaume, Sylvain Kahane & Guy Perrier. 2018. SUD or surface-syntactic universal dependencies: An annotation scheme near-isomorphic to UD. In *Proceedings of the second workshop on universal dependencies (UDW 2018)*, 66–74. Brussels: Association for Computational Linguistics.

Gerdes, Kim, Bruno Guillaume, Sylvain Kahane & Guy Perrier. 2019. Improving surface-syntactic universal dependencies (SUD): Surface-syntactic relations and deep syntactic features. In *Proceedings of the 18th international workshop on treebanks & linguistic theory*, 126–132. Paris: Association for Computational Linguistics.

Gibson, Edward. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition* 68(1). 1–76.

Gibson, Edward, Richard Futrell, Steven T. Piantadosi, Isabelle Dautriche, Kyle Mahowald, Leon Bergen & Roger Levy. 2019. How efficiency shapes human language. *Trends in Cognitive Sciences* 23(5). 389–407.

Gildea, Daniel & David Temperley. 2007. Optimizing grammars for minimum dependency length. In *Proceedings of the 45th annual meeting of the association for computational linguistics*, 184–191. Prague: Association for Computational Linguistics.

Gildea, Daniel & David Temperley. 2010. Do grammars minimize dependency length? *Cognitive Science* 34(2). 286–310.

Gómez-Rodríguez, Carlos, Morten H. Christiansen & Ramon Ferrer-i-Cancho. 2019. *Memory limitations are hidden in grammar*. *CoRR* abs/1908.06629.

Gómez-Rodríguez, Carlos, Marco Kuhlmann & Giorgio Satta. 2010. Efficient parsing of well-nested linear context-free rewriting systems. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics, Mexico*, 276–284.

Greenberg, Joseph H. 1963. Some universals of grammar with particular reference to the order of meaningful elements. In Joseph H. Greenberg (ed.), *Universals of language*, 73–113. Cambridge, MA: MIT Press.

Harper, Lawrence H., Jr. 1964. Optimal assignments of numbers to vertices. *Journal of the Society for Industrial Applied Mathematics* 12. 131–135.

Haspelmath, Martin. 2008. Parametric versus functional explanations of syntactic universals. In Theresa Biberauer (ed.), *The limits of syntactic variation*, 75–107. Amsterdam: Benjamins.

Havelka, Jiří. 2007. Beyond projectivity: Multilingual evaluation of constraints and measures on non-projective structures. In *Proceedings of the 45th annual meeting of the association for computational linguistics*, 608–615. Prague: Association for Computational Linguistics.

Hawkins, John A. 1994. *A performance theory of order and constituency*. Cambridge: Cambridge University Press.

Hawkins, John A. 2004. *Efficiency and complexity in grammars*. Oxford: Oxford University Press.

Hawkins, John A. 2014. *Cross-linguistic variation and efficiency*. Oxford: Oxford University Press.

Hochberg, Robert A. & Matthias F. Stallmann. 2003. Optimal one-page tree embeddings in linear time. *Information Processing Letters* 87. 59–66.

Hockett, Charles F. 1960. The origin of language. *Scientific American* 203(3). 88–96.

Hopcroft, John E. & Jeffrey D. Ullman. 1979. *Introduction to automata theory, languages and computation*. Boston, MA: Addison-Wesley.

ISO 639-3. 2007. *Codes for the representation of names of languages — Part 3: Alpha-3 code for comprehensive coverage of languages*. Geneva, CH: Standard International Organization for Standardization.

Joshi, Aravind K. 1990. Processing crossed and nested dependencies: An automaton perspective on the psycholinguistic results. *Language and Cognitive Processes* 5. 1–27.

Joshi, Aravind K., Krishnamurti Vijay-Shanker & David J. Weir. 1991. The convergence of mildly context-sensitive grammar formalisms. In Peter Sells, Stuart M. Shieber & Thomas Wasow (eds.), *Foundational issues in natural language processing*, 31–81. Cambridge, MA: MIT Press.

Kuhlmann, Marco. 2013. Mildly non-projective dependency grammar. *Computational Linguistics* 39(2). 355–387.

Kuhlmann, Marco & Joakim Nivre. 2006. Mildly non-projective dependency structures. In *Proceedings of the COLING/ACL 2006 main conference poster sessions*, 507–514. Sydney: Association for Computational Linguistics

Liu, Haitao. 2008. Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science* 9(2). 159–191.

Liu, Haitao. 2010. Dependency direction as a means of word-order typology: A method based on dependency treebanks. *Lingua* 120(6). 1567–1578.

Liu, Haitao, Chunshan Xu & Junying Liang. 2017. Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of Life Reviews* 21. 171–193.

Lu, Qian & Haitao Liu. 2016. A quantitative study of the relationship between crossing and distance in human language. *Journal of Shanxi University (Philosophy & Science)* 39(4). 49–56.

Maier, Wolfgang & Timm Lichte. 2009. Characterizing discontinuity in constituent treebanks. In *International conference on formal grammar*, 167–182.

Mambrini, Francesco & Marco Passarotti. 2013. Non-projectivity in the ancient greek dependency treebank. In *Proceedings of the second international conference on dependency linguistics (DepLing 2013)*, 177–186. Prague: Charles University in Prague.

Marcus, Solomon. 1965. Sur la notion de projectivité. *Mathematical Logic Quarterly* 11(2). 181–192.

Michaelis, Jens. 1998. Derivational minimalism is mildly context-sensitive. In *Logical aspects of computational linguistics*, 179–198. Berlin: Springer.

Miletic, Aleksandra & Assaf Urieli. 2017. Non-projectivity in Serbian: Analysis of formal and linguistic properties. In *Proceedings of the fourth international conference on dependency linguistics (DepLing 2017)*, 135–144. Pisa: Association for Computational Linguistics.

Nivre, Joakim. 2015. Towards a universal grammar for natural language processing. In *Computational linguistics and intelligent text processing*, 3–16. Berlin: Springer.

Nivre, Joakim, et al. 2019. Universal dependencies 2.4. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL). Faculty of Mathematics and Physics, Charles University. Available at: http://hdl.handle.net/11234/1-2988.

Nivre, Joakim & Jens Nilsson. 2005. Pseudo-projective dependency parsing. In *Proceedings of the 43rd annual meeting of the association for computational linguistics*, 99–106. Ann Arbor: Association for Computational Linguistics.

Park, Y. Albert & Roger Levy. 2009. Minimal-length linearizations for mildly context-sensitive dependency trees. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, 335–343. Boulder: Association for Computational Linguistics.

Pitler, Emily, Sampath Kannan & Mitchell Marcus. 2013. Finding optimal 1-endpoint-crossing trees. *Transactions of the Association for Computational Linguistics* 1. 13–24.

Pollard, Carl & Ivan A. Sag. 1994. *Head-driven phrase structure grammar*. Stanford, CA: Center for the Study of Language and Information.

Shieber, Stuart M. 1985. Evidence against the context-freeness of natural language. In *The formal complexity of natural language*, 320–334. Berlin: Springer.

Shiloach, Yossi. 1979. A minimum linear arrangement algorithm for undirected trees. *SIAM Journal on Computing* 8(1). 15–32.

Steedman, Mark & Jason Baldridge. 2011. Combinatory categorial grammar. *Non-transformational syntax: Formal and explicit models of grammar*, 181–224. Oxford: Blackwell Publishing Ltd.

Temperley, David. 2008. Dependency-length minimization in natural and artificial languages. *Journal of Quantitative Linguistics* 15(3). 256–282.
Temperley, David & Daniel Gildea. 2018. Minimizing syntactic dependency lengths: Typological/cognitive universal? *Annual Review of Linguistics* 4. 1–15.
von der Gabelentz, Georg. 1901. *Die Sprachwissenschaft, ihre Aufgaben, Methoden, und bisherigen Ergebnisse*. Leipzig: Weigel.
Weir, David J. 1988. *Characterizing mildly context-sensitive grammar formalisms*. Philadelphia, PA: University of Pennsylvania Dissertation.
Yadav, Himanshu, Samar Husain & Richard Futrell. 2019. Are formal restrictions on crossing dependencies epiphenomenal? In *Proceedings of the 18th international workshop on treebanks & linguistic theory*, 2–12. Paris: Association for Computational Linguistics.
Yadav, Himanshu, Ashwini Vaidya & Samar Husain. 2017. Understanding constraints on non-projectivity using novel measures. In *Proceedings of the fourth international conference on dependency linguistics (DepLing 2017)*, 276–286. Pisa: Association for Computational Linguistics.
Yadav, Himanshu, Ashwini Vaidya, Vishakha Shukla & Samar Husain. 2020. Word order typology interacts with linguistic complexity: A cross-linguistic corpus study. *Cognitive Science* 44(4). e12822.
Yan, Jianwei & Haitao Liu. 2019. Which annotation scheme is more expedient to measure syntactic difficulty and cognitive demand? In *Proceedings of the first workshop on quantitative syntax (Quasy, SyntaxFest 2019)*, 16–24. Paris: Association for Computational Linguistics.
Zipf, George Kingsley. 1949. *Human behavior and the principle of least effort*. Oxford, UK: Addison-Wesley Press.

Supplementary Material: The online version of this article offers supplementary material (https://doi.org/10.1515/lingvan-2019-0070).

Do dependency lengths explain constraints on crossing dependencies?

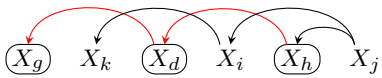
Supplementary Materials

S1. Method for generating random trees

In order to sample a random tree structure from the uniform distribution over tree structures, we use the method of Prüfer codes. In graph theory, every directed tree of length n can be identified uniquely with a sequence of $n-1$ natural numbers in the range $\{1, \ldots, n\}$ (Prüfer, 1918). These sequences are called Prüfer codes. Therefore, we can sample a random tree of length n by first sampling a random Prüfer code, and then converting the Prüfer code into a tree structure using the algorithm given by Nijenhuis and Wilf (1978, Ch. 28).

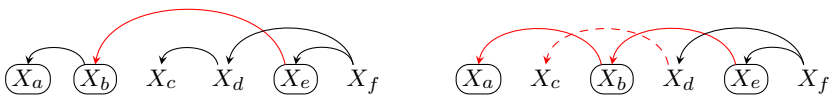
S2. Formal definitions of crossing constraints

See Figures 1, 2, 3, and 4.



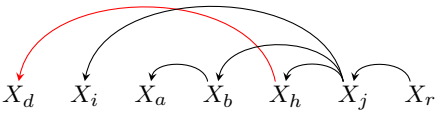
Gap degree 2

Fig. 1: Definition of gap degree. The **full projection** of a node X is the set of all the nodes transitively dominated by X plus X itself. For example, in the dependency tree above, $\{X_g, X_d, X_h\}$ is the projection of the node X_h. A projection has a **gap** at each point where there are one or more nodes X intervening between two nodes X_i, X_j of the full projection where X are not transitive descendants of either X_i or X_j. For example, the full projection of X_j, i.e. $\{X_h, X_d, X_g\}$, has two gaps, one between X_h and X_d, and another between X_d and X_g. The **gap degree** of a tree is the largest number of gaps in the full projection of any node in the tree. The gap degree of the tree above is 2.

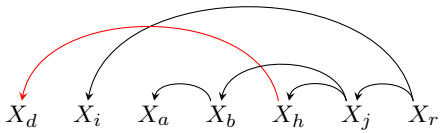


(a) Well-nested (b) Ill-nested

Fig. 2: Definition of well-nestedness. The **subtree** rooted at a node X is the set of all the transitive nodes dominated by X. For example, in the dependency tree (a) and (b) above, $\{X_a, X_b, X_e\}$ is the subtree rooted at node X_e, and $\{X_c, X_d\}$ is the subtree rooted at node X_d. Two subtrees with nodes $\{P, Q\}$ and $\{R, S\}$ **interleave** if the nodes are in linear order such that $P < R < Q < S$. A dependency tree is **ill-nested** iff two of its disjoint subtrees interleave. For example, in (a), $\{X_a, X_b, X_e\}$ and $\{X_c, X_d\}$ are two disjoint subtrees but they do not interleave as the nodes are in the order $X_a < X_b < X_c < X_d < X_e$. Therefore, tree (a) is well-nested. In (b), the disjoint subtrees $\{X_a, X_b, X_e\}$ and $\{X_c, X_d\}$ interleave as the order of the nodes is $X_a < X_c < X_b < X_d < X_e$. The dashed red arc creates the ill-nestedness. Ill-nestedness implies gap degree ≥ 1.



(a) Edge degree 2, End-point crossings 1



(b) Edge degree 2, End-point crossings 2

Fig. 3: Definition of edge degree and end-point crossings. Let e be the **span** of dependency arc $X_h \rightarrow X_d$, consisting of nodes between a head X_h and its dependent X_d, which are X_i, X_a, and X_b in both trees above. We call a node in e an **intervener** iff it is not a transitive descendant of X_h nor of any other node in e. The **edge degree** of a dependency arc $X_h \rightarrow X_d$ is the number of interveners in the span e. The number of **end-point crossings** for an arc $X_h \rightarrow X_d$ is the number of interveners in the span e *with distinct heads*. For example, the arc $X_h \rightarrow X_d$ in both **(a)** and **(b)** has an edge degree of 2 because it has two interveners X_i and X_b in both cases. But in **(a)**, the arc $X_h \rightarrow X_d$ has end-point crossings 1 because two the interveners X_i and X_b share a head X_j, whereas in **(b)**, it has end-point crossings 2 because the two interveners X_i and X_b have two distinct heads, namely X_j and X_r respectively. The edge degree of a tree is the highest edge degree among any of its arcs; likewise for end-point crossings.

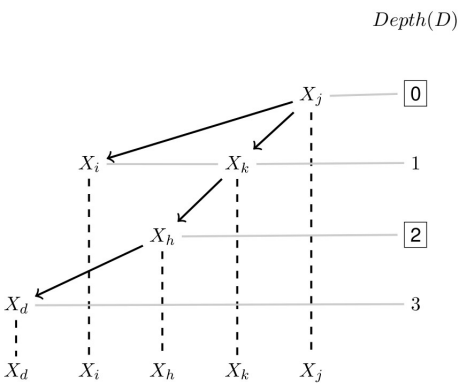


Fig. 4: The **heads' depth difference** (HDD) for a crossing arc $X_h \rightarrow X_d$ with intervener X_i is defined as the difference between the depth of X_h and the depth of the head of the intervener X_j: $\text{HDD}(X_h, X_d) = \text{depth}(X_h) - \text{depth}(X_j)$. The HDD of a dependency tree is the maximum HDD among the HDDs of the arcs and interveners in the tree. If there are no crossing arcs, HDD of the tree is zero.

S3. Model for the rate of crossing dependencies

To predict rate of crossing dependencies as function of sentence length, we fit a Poisson regression model:

$$\log N_i = \beta_0 + \beta_l \log S_i + \beta_r 1_{\text{real}} + \beta_{rl} \log S_i 1_{\text{real}} + \epsilon_i, \tag{1}$$

where N_i is the number of crossing dependencies in i^{th} sentence. $\log S_i$ is the log-length of the i^{th} sentence. 1_{real} is an indicator variable with value 1 for a real tree and 0 for a baseline tree. The regression is equivalent to modeling the number of crossings N_i as a power law

$$N_i = cS_i^{\alpha},$$

with $c = e^{\beta_0}$ and $\alpha = \beta_l$ for baseline trees, and $c = e^{\beta_0 + \beta_r}$ and $\alpha = \beta_l + \beta_{rl}$ for real trees. To see this, we note that the regression equation in Eq. 1 can be rewritten as a power law by exponentiating both sides and using the identity $e^{a \log b} = b^a$:

$$\begin{aligned} N_i &= e^{\beta_0} e^{\beta_l \log S_i + \beta_r 1_{\text{real}} + \beta_{rl} \log S_i 1_{\text{real}}} \\ &= e^{\beta_0 + \beta_r 1_{\text{real}}} S_i^{\beta_l + \beta_{rl} 1_{\text{real}}} \\ &= cS_i^{\alpha}. \end{aligned} \tag{2}$$

We found that Poisson regressions have the best log-likelihood when predicting our data, as compared with linear regressions and quadratic regressions.

We evaluate whether there is a significant difference between real and random trees by checking whether a regression as in Eq. 1 is a significantly better fit to the data than regression in Eq. 3:

$$\log N_i = \beta_0 + \beta_l \log S_i + \epsilon_i. \tag{3}$$

The regression in Eq. 3 lacks the 1_{real} terms and therefore has no means to distinguish between real and baseline trees. We test whether full model provides a significantly better fit than Eq. 3 using a χ^2 test on the likelihood ratio.

This test showed that the rate of crossing dependencies was significantly different in real trees compared to DL-controlled random trees and DL-controlled RLAs. See Table 1 for full results.

We note that the significance and sign of the difference between real and baseline trees is the same regardless of regression method used (linear regression, quadratic regression, Poisson regression with untransformed sentence length, Poisson regression with log-transformed sentence length). Poisson regression with log-transformed sentence length had the best fit to the data according to data likelihood.

Baseline	Estimate	Standard Error	z value	p-value
DL-controlled random trees baseline				
Intercept	-4.63	0.01	-233.83	<0.001 *
log(s.length)	2.47	0.01	254.85	<0.001 *
Observed	-1.80	0.04	-36.18	<0.001 *
log(s.length):Observed	0.08	0.02	3.43	<0.001 *
DL-controlled random linear arrangements				
Intercept	-4.64	0.02	-202.95	<0.001 *
log(s.length)	2.31	0.01	215.71	<0.001 *
Observed	-0.92	0.05	-18.93	<0.001 *
log(s.length):Observed	-0.19	0.02	-8.37	<0.001 *

Tab. 1: This table shows the fitted estimates from model 1 for DL-controlled random trees baseline and DL-controlled Random Linear Arrangements.

S4. Model for the crossing constraints

To predict rate of crossing constraint violation (e.g., gap degree) as function of sentence length or tree depth or tree arity, we fit mixed-effect Poisson regressions with varying intercepts for languages. For example, following regression is fitted to predict gap degree as a function of sentence length:

$$\log G_i = \beta_0 + u_{0j} + \beta_l S_i + \beta_r 1_{\text{real}} + \beta_{rl} S_i 1_{\text{real}} + \epsilon_i, \tag{4}$$

where G_i is the gap degree of i^{th} sentence. S_i is the length of the i^{th} sentence. S_i has been centered here. 1_{real} is an indicator variable with value 1 for a real tree and 0 for a baseline tree. u_{0j} is the random intercept adjustment for j^{th} language.

Note that our dependent variable can be gap degree, well-nestedness, edge degree, endpoint crossing or HDD. The predictor variable can be sentence length, tree depth, arity or sum dependency length. Given all these combinations, we fit 20 models in total for both random trees baseline and RLAs which controls for both DL-sequence and rate of crossing dependencies. Table 2 provides estimates for all crossing constraints given different predictors for random trees baseline and RLAs controlled for DL and crossings rate. Note that our critical test for significance in the main paper is not based on any single coefficient from this table, but rather is based on a likelihood ratio test comparing models with and without the coefficients that distinguish real from baseline trees.

Dependent variable	Independent variable	DL-controlled random trees β Estimate	Std. Error	p value		DL-controlled RLAs β Estimate	Std. Error	p value	
Gap degree	S. length	1.00	0.008	<2e-16	*	0.63	0.007	<2e-16	*
	Observed	-0.04	0.014	0.001	*	-0.004	0.011	0.709	n.s.
	S. length × Observed	-0.006	0.011	0.572	n.s.	-0.004	0.009	0.689	n.s.
	Arity	0.18	0.007	<2e-16	*	0.02	0.006	4.31e-05	*
	Observed	-0.10	0.010	<2e-16	*	-0.007	0.009	0.427	n.s.
	Arity × Observed	-0.01	0.009	0.08	n.s.	-0.007	0.009	0.400	n.s.
	Depth	0.78	0.005	<2e-16	*	0.56	0.005	<2e-16	*
	Observed	0.16	0.013	<2e-16	*	-0.004	0.010	0.673	n.s.
	Depth × Observed	0.05	0.008	2.88e-10	*	-0.003	0.007	0.621	n.s.
	$\sum$ DL	0.62	0.003	<2e-16	*	0.34	0.002	<2e-16	*
	Observed	0.07	0.011	1.2e-09	*	-0.005	0.009	0.554	n.s.
	$\sum$ DL × Observed	-0.17	0.004	<2e-16	*	-0.006	0.003	0.021	*
Well-nestedness	S. length	1.35	0.018	<2e-16	*	0.91	0.013	<2e-16	*
	Observed	0.27	0.035	6.1e-15	*	0.02	0.023	0.358	n.s.
	S. length × Observed	-0.06	0.024	0.006	*	0.008	0.019	0.644	n.s.
	Arity	0.43	0.014	<2e-16	*	0.34	0.010	<2e-16	*
	Observed	0.02	0.022	0.297	n.s.	0.01	0.017	0.335	n.s.
	Arity × Observed	0.04	0.018	0.014	*	0.02	0.011	0.100	n.s.
	Depth	0.63	0.012	<2e-16	*	0.39	0.010	<2e-16	*
	Observed	0.36	0.025	<2e-16	*	0.02	0.018	0.264	n.s.
	Depth × Observed	0.04	0.017	0.010	*	0.01	0.014	0.303	n.s.
	$\sum$ DL	0.79	0.007	<2e-16	*	0.39	0.003	<2e-16	*
	Observed	0.46	0.025	<2e-16	*	0.02	0.017	0.267	n.s.
	$\sum$ DL × Observed	-0.27	0.007	<2e-16	*	0.001	0.003	0.901	n.s.
Edge degree	S. length	0.69	0.006	<2e-16	*	0.42	0.005	<2e-16	*
	Observed	-0.02	0.011	0.047	*	-0.02	0.008	0.005	*
	S. length × Observed	-0.02	0.009	0.005	*	-0.02	0.008	0.002	*
	Arity	-0.002	0.006	0.703	n.s.	-0.02	0.005	6.61e-06	*
	Observed	-0.04	0.008	4.61e-08	*	-0.03	0.007	7.64e-06	*
	Arity × Observed	0.05	0.008	6.21e-09	*	-0.01	0.007	0.192	n.s.
	Depth	0.56	0.005	<2e-16	*	0.41	0.004	<2e-16	*
	Observed	0.15	0.008	0.0002	*	-0.02	0.008	0.0009	*
	Depth × Observed	0.01	0.007	0.266	n.s.	-0.01	0.006	0.0438	*
End-point crossing	S. length	0.68	0.006	<2e-16	*	0.41	0.005	<2e-16	*
	Observed	-0.03	0.011	0.002	*	-0.03	0.008	0.0007	*
	S. length × Observed	-0.03	0.009	8.6e-05	*	-0.03	0.008	0.0002	*
	Arity	-0.02	0.006	0.009	*	-0.06	0.005	<2e-16	*
	Observed	-0.05	0.008	6.80e-11	*	-0.04	0.007	8.63e-08	*
	Arity × Observed	0.03	0.008	6.27e-05	*	-0.01	0.007	0.216	n.s.
	Depth	0.56	0.005	<2e-16	*	0.41	0.004	<2e-16	*
	Observed	0.13	0.010	<2e-16	*	-0.03	0.008	5.04e-05	*
	Depth × Observed	0.009	0.007	0.207	n.s.	-0.01	0.006	0.046	*
HDD	S. length	0.68	0.004	<2e-16	*	0.40	0.004	<2e-16	*
	Observed	-0.15	0.007	<2e-16	*	-0.09	0.006	<2e-16	*
	S. length × Observed	-0.05	0.006	1.2e-14	*	-0.04	0.005	2.51e-12	*
	Arity	-0.08	0.004	<2e-16	*	-0.14	0.004	<2e-16	*
	Observed	-0.17	0.006	<2e-16	*	-0.11	0.005	<2e-16	*
	Arity × Observed	0.03	0.006	1.02e-06	*	-0.002	0.005	0.695	n.s.
	Depth	0.65	0.003	<2e-16	*	0.49	0.003	<2e-16	*
	Observed	0.04	0.007	2.59e-10	*	-0.08	0.006	<2e-16	*
	Depth × Observed	-0.01	0.005	0.011	*	-0.03	0.004	3.76e-13	*

Tab. 2: Mixed-effect Poisson regression results for all the crossing constraints and dependency tree measures. "Observed" is an indicator variable with value 1 for observed trees and 0 for random trees, the same as 1_{real} in Equation 4. We do not directly interpret the fitted estimates for testing our null hypothesis. We compare the models with and without 1_{real} terms using likelihood ratio to test whether there is significant difference between real and random baseline trees. See previous section for more details.

References

Nijenhuis, A. and Wilf, H. S. (1978). *Combinatorial Algorithms*. Computer Science and Applied Mathematics. Academic Press, New York, 2 edition.

Prüfer, H. (1918). Neuer Beweis eines Satzes über Permutationen. *Archiv der Mathematischen Physik*, 27:742 – 744.