Memory limitations are hidden in grammar

Carlos Gómez-Rodríguez¹, Morten H. Christiansen²,³ and Ramon Ferrer-i-Cancho⁴∗

¹FASTPARSE Lab, LyS Research Group, Departamento de Computación, Universidade da Coruña, CITIC, Campus de Elviña s/n, 15071, A Coruña, Spain
²Department of Psychology, Cornell University, Ithaca, NY 14853, USA
³Interacting Minds Centre and School of Communication and Culture, Nobelparken, Aarhus University, 8000 Aarhus C, Denmark
⁴Complexity and Quantitative Linguistics Lab, LARCA Research Group Departament de Ciències de la Computació, Universitat Politècnica de Catalunya (UPC) Jordi Girona 1-2, 08034, Barcelona, Spain

∗To whom correspondence should be addressed; E-mail: rferrericancho@cs.upc.edu.

The ability to produce and understand an unlimited number of different sentences is a hallmark of human language. Linguists have sought to define the essence of this generative capacity using formal grammars that describe the syntactic dependencies between constituents, independent of the computational limitations of the human brain. Here, we evaluate this independence assumption by sampling sentences uniformly from the space of possible syntactic structures. We find that the average dependency distance between syntactically related words, a proxy for memory limitations, is less than expected by chance in a collection of state-of-the-art classes of dependency grammars. Our findings indicate that memory limitations have permeated grammatical descriptions, suggesting that it may be impossible to build a parsimonious theory of human linguistic productivity independent of non-linguistic cognitive
Analyses of billions of sentences reveal that human linguistic productivity cannot be captured without cognitive constraints.

An often celebrated aspect of human language is its capacity to produce an unbounded number of different sentences (1, 2). For many centuries, the goal of linguistics has been to capture this capacity by a formal description—a grammar—consisting of a systematic set of rules and/or principles that determine which sentences are part of a given language and which are not (3). Over the years, these formal grammars have taken many forms but common to them all is the assumption that they capture the idealized linguistic competence of a native speaker/hearer, independent of any memory limitations or other non-linguistic cognitive constraints (1, 2). These abstract formal descriptions have come to play a foundational role in the language sciences, from linguistics, psycholinguistics, and neurolinguistics (4, 5) to computer science, engineering, and machine learning (6–8). Despite evidence that processing difficulty underpins the unacceptability of certain sentences (9, 10), the cognitive independence assumption that is a defining feature of linguistic competence has not been examined in a systematic way using the tools of formal grammar. It is therefore unclear whether these supposedly idealized descriptions of language are free of non-linguistic cognitive constraints, such as memory limitations.

If the cognitive independence assumption should turn out not to hold, then it would have wide-spread theoretical and practical implications for our understanding of human linguistic productivity. It would require a reappraisal of key parts of linguistic theory that hitherto have posed formidable challenges for explanations of language processing, acquisition and evolu-
You brought your dog. Someone arrived with red hair. A hearing is scheduled on the issue today. A mildly non-projective tree from the classes $1EC$ and $MH_4$ (adapted from (15)) where $n = 8$ and $\langle d \rangle = 13/7 \approx 1.85$. B. A planar but non-projective tree where $n = 5$ and $\langle d \rangle = 3/2$ (adapted from (16)). C. A projective tree (adapted from (16)) where $n = 4$ and $\langle d \rangle = 4/3$. D. A diagram of the superset relationships between projective, planar, mildly non-projective and unrestricted (all) syntactic dependency structures.

In dependency grammar, the syntactic structure of a sentence is defined by two components. First, a directed graph where vertices are words and arcs indicate syntactic dependencies be-
tween a head and its dependent. Such a graph has a root (a vertex that receives no edges) and edges are oriented away from the root (Figure 1). Second, the linear arrangement of the vertices of the graph (defined by the sequential order of the words in a sentence). Thus, syntactic dependency structures constitute a particular kind of spatial network where the graph is embedded in one dimension (17), a correspondence that has led to the development of syntactic theory from a network theory standpoint (13).

Dependency grammar is an important framework for various reasons. First, categorial grammar defines the syntactic structure of a sentence as dependency grammar (9). Second, equivalences exist between certain formalisms of dependency grammar and constituency grammar (18, 19). Third, there has been an evolution of minimalism towards dependency grammar (20). Finally, dependency grammar has become a de facto standard in computational linguistics (21).

To delimit the set of possible grammatical descriptions, various classes or sets of syntactic dependency structures have been proposed. These classes can be seen as filters on the possible linear arrangements of a given tree. Here, we consider four main classes. First, consider planar structures, where edges do not cross when drawn above the words of the sentence. The structure in Figures 1B-C are planar while that of Figure 1A is not. Second, we have projective structures, the most well-known class. A dependency tree is projective if, and only if, it is planar and its root is not covered by any dependency (Figure 1C). Third, there are mildly non-projective structures, comprising the union of planar structures and additional structures with further (but slight) deviations from projectivity, e.g., by having a low number of edge crossings (Figure 1A). Finally, the class of all structures, that has no constraints on the possible structures.

Fig. 1D shows the inclusion relationships among these classes. However, the whole picture, encompassing state-of-the-art classes is more complex. Mildly non-projective structures are not actually a class but a family of classes. We have selected three representative classes: \( MH_k \),
$WG_1$ and $1EC$ structures, that are supersets of projective structures but whose definition is more complex (Supplementary Text).

Here we validate the assumption of independence between grammatical constraints and cognitive limitations in these classes of grammar using the distance between syntactically related words in a dependency tree as a proxy for memory constraints (22, 23). Such a distance is defined as the number of intermediate words plus one. Thus, if the linked words are consecutive they are at distance 1, if they are separated by an intermediate word they are at distance two, and so on, as shown in Figure 1. Dependency distance minimization, a type of memory constraint, is believed to result from pressure against decay of activation or interference during the processing of sentences (22, 23). Dependency distances reduce in case of cognitive impairment (24).

Assuming that all the linear arrangements are equally likely, $\langle d \rangle$, the average of dependency distances in a sentence of $n$ words, is expected to be (25)

$$\langle d \rangle_{rla} = \frac{n + 1}{3}. \quad (1)$$

Figure 2A shows that $\langle d \rangle_{RS}$, the average dependency distance in attested syntactic dependency structures (RS), is below the random baseline defined by $\langle d \rangle_{rla}$ (see Supplementary Text for a justification of this baseline). This is in line with previous statistical analyses (25–28) (see (22, 23) for a broader review of previous work) and the expected influence of performance constraints on attested trees.

The fact that $\langle d \rangle_{RS}$ is below 4 has been interpreted as a sign that dependency lengths are constrained by working memory limitations (26). For this reason, we test whether memory effects have permeated the classes of grammar by determining if $\langle d \rangle_{AS}$, the average dependency distance in a collection of artificial syntactic dependency structures (AS) from a certain class, is also below $\langle d \rangle_{rla}$. The purpose of Figure 2A is merely to provide the reader with a baseline derived from attested dependency structures in natural language as a backdrop for the main
The contribution of the article, which is based on artificial syntactic dependency structures.

For a given $n$, we generate an ensemble of artificial syntactic dependency structures by exhaustive sampling for $n \leq n^* = 10$ and random sampling for $n > n^*$ (Supplementary Materials). These artificial syntactic dependency trees are only constrained by the definition of the different classes. They are thus free from any memory constraint other than the ones the different classes of grammars may, perhaps, impose indirectly. Still, these artificial syntactic structures have dependency lengths that are below the chance level (Figure 2B), indicating that memory constraints are hidden in the definition of these classes. Interestingly, $\langle d \rangle_{\text{AS}}$ is below chance for sufficiently large $n$ in all classes of grammars although $\langle d \rangle_{\text{AS}}$ could be above $\langle d \rangle_{\text{rla}}$ in principle (Supplementary Text). In general, the largest reduction of $\langle d \rangle_{\text{AS}}$ with respect to the random baseline is achieved by the projective class, followed by the planar class.

It is worth noting that a reduction of $\langle d \rangle_{\text{AS}}$ with respect to our random baseline has been observed for the projective class in past work, but with some important caveats: (26) did not control for sentence length as in Figure 2B; and whereas (27) did implement this control and considered another class of marginal interest (2-component structures) in addition to projective trees, their use of attested dependency trees instead of artificial control trees suggests that memory limitations might have influenced the results.

The reduction of $\langle d \rangle$ with respect to the random baseline in artificial trees from a wide range of state-of-the-art classes is consistent with the hypothesis that the scarcity of crossing dependencies is a side-effect of pressure to reduce the distance between syntactically related words (13). The smaller reduction of dependency distances with respect to the random baseline in artificial dependency structures can be explained by the fact that the curves in Figure 2B derive from uniform sampling of the space of all possible trees. In contrast, real speakers may not only choose linear arrangements that reduce dependency distance, but also sample the space of possible structures with a bias towards structures that facilitate that such reduction or
Figure 2: The average dependency length, $\langle d \rangle$, as a function of $n$, the sentence length (in words). For reference, the baseline defined by a random linear arrangement of the words of the sentence, $\langle d \rangle_{rla}$ is also shown (dashed line). A. Attested syntactic dependency trees (RS) following three different annotation criteria: UD, Prague and Stanford dependencies. B. Artificial syntactic dependency structures (AS) belonging to different classes of grammars. Due to undersampling, only points represented by at least 30 structures are shown for $n > n^*$. 
that satisfy other cognitive constraints (29).

Our findings complete our understanding of the relationship between projectivity or mildly non-projectivity and dependency distance minimization. It has been shown that such minimization leads to a number of edge crossings that is practically zero (30), and to not covering the root (one of the conditions for projectivity, in addition to planarity) (31). Here, we have demonstrated a complementary effect, i.e., that dependency distance reduces below chance when edge crossings are minimized (planarity) or projectivity is imposed. Whereas a recent study of similar classes of grammars suggested that crossing dependencies are constrained by either grammar or cognitive pressures rather than occurring naturally at some rate —

citeYadav2019a, our findings strongly demonstrate that it is not grammar but rather non-linguistic cognitive constraints, that limit the occurrence of crossing dependencies in languages.

We sampled about 16 billion syntactic dependency structures, that differed in length and syntactic complexity, to determine whether linguistic grammars are free of non-linguistic cognitive constraints, as is typically assumed. Strikingly, while previous work on natural languages has shown that dependency lengths are considerably below what would be expected by a random baseline without memory constraints (25–27,32), we still observe a drop in dependency lengths for randomly generated, mildly non-projective structures that supposedly abstract away from cognitive limitations. Our interpretation of these results is that memory constraints, in the form of dependency distance minimization, have become inherent to formal linguistic grammars.

It may be objected that our conclusions are limited by the sample of classes that we have considered and that we cannot not exclude the possibility that, in the future, researchers might adopt a new class of mildly non-projective structures whose dependency distances cannot be distinguished from the random baseline. However, we believe that this is very unlikely for the following reasons: (1) our current sample of classes is representative of the state of the
art (33), and spans classes that originated with different goals and motivations (from purely theoretical to parsing efficiency), with all sharing the drop in dependency lengths, (2) while one could conceivably engineer a class to have lengths in line with the baseline while still having high coverage of linguistic phenomena, this would mean forwarding more responsibility for dependency distance reduction to other parts of the linguistic theory in order to warrant that dependency distances are reduced to a realistic degree (Fig. 2) and hence would preclude a parsimonious approach to language, and (3) given the positive correlation between crossings and dependency lengths (34, 35), such a class would be likely to have many dependency crossings, so it would be, at the least, questionable to call it mildly non-projective.

Beyond upending longheld assumptions about the nature of human linguistic productivity, our findings also have key implications for debates on how children learn language, how language evolved, and how computers might best master language. Whereas a common assumption in the acquisition literature is that children come to the task of language learning with built-in linguistic constraints on what they learn (5, 11), our results suggest that language-specific constraints may not be needed and instead be replaced by general cognitive constraints (36). The strong effects of memory on dependence distance minimization provide further support for the notion that language evolved through processes of cultural evolution shaped by the human brain (12), rather than the biological evolution of language-specific constraints (5). Finally, our results raise the intriguing possibility that if we want to develop computer systems that target human linguistic ability in the context of human-computer interaction, we may paradoxically need to constraint the power of such systems to be in line with human cognitive limitations, rather than giving them the super-human computational capacity of AlphaGo.

Our study was conducted using the framework of dependency grammar, but because of the close relationship between this framework and other ways of characterizing the human unbounded capacity, such as categorial grammar (9), phrase structure grammar (18, 19), and
minimalist grammar (20), our results suggest that any parsimonious grammatical framework will incorporate memory constraints. Moreover, given that dependency grammars constitute a special case of a graph that is embedded in one dimension, the physics toolbox associated with statistical mechanics and network theory may be applied to provide further insight into the nature of human linguistic productivity (13, 17). However, these future explorations notwithstanding, our current findings show that memory limitations have permeated current linguistic conceptions of grammar, suggesting that it may not be possible to adequately capture our unbounded capacity for language, at least in the context of a parsimonious theory compatible with the idea of mild non-projectivity, without incorporating non-linguistic cognitive constraints into the grammar formalism.

References


Acknowledgments

We thank two anonymous reviewers from Science for their many suggestions to improve the article. We are also to L. Alemany-Puig, A. Hernandez-Fernandez and M. Vitevitch for helpful comments. Funding: CGR has received funding from the European Research Council (ERC), under the European Union’s Horizon 2020 research and innovation programme (FAST-PARSE, grant agreement No 714150), from the ANSWER-ASAP project (TIN2017-85160-C2-1-R) from MINECO (Ministerio de Economia, Industria y Competitividad), and from Xunta de Galicia (ED431B 2017/01, and an Oportunius program grant to complement ERC grants). RFC is supported by the grant TIN2017-89244-R from MINECO and the recognition 2017SGR-856 (MACDA) from AGAUR (Generalitat de Catalunya). Author contributions: R.F.C and C.G.R designed the experiments. C.G.R. performed the experiments. R.F.C., M.H.C. and C.G.R wrote the article. Competing interests: Authors declare no competing interest. Data and materials availability: All data are available either as auxiliary files published with the article or in open access repositories indicated in the supplementary materials.

Materials and Methods

Materials

We estimated the average dependency distances in attested sentences using collections of syntactic dependency treebanks from different languages. A syntactic dependency treebank is a database of sentences and their syntactic dependency trees.

To provide results on a wide range of languages while controlling for the effects of different syntactic annotation theories, we use two collections of treebanks:

- Universal Dependencies (UD), version 2.4 (37). This is the largest available collection of syntactic dependency treebanks, featuring 146 treebanks from 83 distinct languages. All
of these treebanks are annotated following the common Universal Dependencies annotation criteria, which are a variant of the Stanford Dependencies for English (38), based on lexical-functional grammar (39), adapting them to be able to represent syntactic phenomena in diverse languages under a common framework. This collection of treebanks can be freely downloaded\[1\] and is available under free licenses.

- HamleDT 2.0 (40). This collection is smaller than UD, featuring 30 languages, all of which (except for one: Bengali) are also available in UD, often with overlapping source material. Thus, using this collection does not meaningfully extend the diversity of languages covered beyond using only UD. However, the interest of HamleDT 2.0 lies in that each of the 30 treebanks is annotated with not one, but two different sets of annotation criteria: Universal Stanford dependencies (41) and Prague Dependencies (42). We abbreviate these two subsets of the HamleDT 2.0 collection as “Stanford” and “Prague”, respectively. While Universal Stanford dependencies are closely related to UD, Prague dependencies provide a significantly different view of syntax, as they are based on the functional generative description (43) of the Praguian linguistic tradition (44), which differs from Stanford dependencies in substantial ways, like the annotation of conjunctions or adpositions (45). Thus, using this version of HamleDT\[2\] makes our analysis more robust, as we can draw conclusions without being tied to a single linguistic tradition. The HamleDT 2.0 treebanks are available online\[3\]. While not all of the treebanks are made fully available to the public under free licenses, to reproduce our analysis it is sufficient to use a stripped version where the words have been removed from the sentences for licensing reasons, but the bare trees are available. This version is distributed freely\[4\].

\[1\]https://universaldependencies.org/
\[2\]While there is a later version (HamleDT 3.0), it abandoned the dual annotation and adopted Universal Dependencies instead, thus making it less useful for our purposes.
\[3\]https://ufal.mff.cuni.cz/hamledt/hamledt-treebanks-20
\[4\]https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/
A preprocessed file with the minimal information needed to reproduce Figure 1A is available.

**Methods**

In our study, we do not investigate the average dependency distance over a whole ensemble of dependency structures but conditioning on sentence length \( (32) \). Then for a given \( n \), we calculate \( \langle d \rangle_{AS} \), the average dependency length for an ensemble of artificial syntactic dependency structures (AS), and also \( \langle d \rangle_{RS} \), the average dependency length for an ensemble of attested syntactic dependency structures (RS). By doing that, we are controlling for sentence length, getting rid of the possible influence of the distribution of sentence length in the calculation of \( \langle d \rangle_{RS} \) or \( \langle d \rangle_{AS} \) \( (32) \).

To preprocess the treebanks for our analysis, we removed punctuation, following common practice in statistical research of dependency structures \( (13) \). We also removed tree nodes that do not correspond to actual words, such as the null elements in the Bengali, Hindi and Telugu HamleDT corpora and the empty nodes in several UD treebanks. To ensure that the dependency structures are still valid trees after these removals, we reattached nodes whose head has been deleted as dependents of their nearest non-deleted ancestor. Finally, in our analysis we disregarded syntactic trees with less than three nodes, as their statistical properties are trivial and provide no useful information (a single-node dependency tree has no dependencies at all, and a 2-node tree always has a single dependency of distance 1). Table 1 summarizes the languages in each collection of treebanks.

Apart from the attested trees, we used a collection of over 16 billion randomly-generated trees. For values of \( n \) (the length or number of nodes) from 3 to \( n^* = 10 \), we exhaustively obtained all possible trees. The number of possible dependency trees for a given length \( n \) is

https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/XHRIYX
Table 1: The languages in every collection grouped by family. The counts attached to the collection names indicate the number of different families and the number of different languages. The counts attached to family names indicate the number of different languages.

<table>
<thead>
<tr>
<th>Collection</th>
<th>Family</th>
<th>Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>UD (19, 83)</td>
<td>Afro-Asiatic (7)</td>
<td>Akkadian, Amharic, Arabic, Assyrian, Coptic, Hebrew, Maltese</td>
</tr>
<tr>
<td></td>
<td>Altaic (3)</td>
<td>Kazakh, Turkish, Uyghur</td>
</tr>
<tr>
<td></td>
<td>Austro-Asiatic (1)</td>
<td>Vietnamese</td>
</tr>
<tr>
<td></td>
<td>Austro-Asiatic (1)</td>
<td>Vietnamese</td>
</tr>
<tr>
<td></td>
<td>Basque (1)</td>
<td>Basque</td>
</tr>
<tr>
<td></td>
<td>Dravidian (2)</td>
<td>Tamil, Telugu</td>
</tr>
<tr>
<td></td>
<td>Indo-European (46)</td>
<td>Afrikaans, Ancient Greek, Armenian, Belarusian, Breton, Bulgarian,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Catalán, Croatian, Czech, Danish, Dutch, English, Faroese, French,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Galician, German, Gothic, Greek, Hindi, Hindi-English, Irish, Italian,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Kurmanji, Latin, Latvian, Lithuanian, Marathi, Norwegian, Old Church</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slavonic, Old French, Old Russian, Persian, Polish, Portuguese, Romanian,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Russian, Sanskrit, Serbian, Slovak, Slovenian, Spanish, Swedish, Ukrainian,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Upper Sorbian, Urdu, Welsh</td>
</tr>
<tr>
<td></td>
<td>Japanese (1)</td>
<td>Japanese</td>
</tr>
<tr>
<td></td>
<td>Korean (1)</td>
<td>Korean</td>
</tr>
<tr>
<td></td>
<td>Mande (1)</td>
<td>Bambara</td>
</tr>
<tr>
<td></td>
<td>Mongolic (1)</td>
<td>Buryat</td>
</tr>
<tr>
<td></td>
<td>Niger-Congo (2)</td>
<td>Wolof, Yoruba</td>
</tr>
<tr>
<td></td>
<td>Other (1)</td>
<td>Naija</td>
</tr>
<tr>
<td></td>
<td>Pama-Nyungan (1)</td>
<td>Warlpiri</td>
</tr>
<tr>
<td></td>
<td>Sign Language (1)</td>
<td>Swedish Sign Language</td>
</tr>
<tr>
<td></td>
<td>Sino-Tibetian (3)</td>
<td>Cantonese, Chinese, Classical Chinese</td>
</tr>
<tr>
<td></td>
<td>Tai-Kadai (1)</td>
<td>Thai</td>
</tr>
<tr>
<td></td>
<td>Tupian (1)</td>
<td>Mbya Guaraní</td>
</tr>
<tr>
<td></td>
<td>Uralic (7)</td>
<td>Erzya, Estonian, Finnish, Hungarian, Karelian, Komi Zyrian, North Sami</td>
</tr>
<tr>
<td>Stanford (7, 30)</td>
<td>Afro-Asiatic (1)</td>
<td>Arabic</td>
</tr>
<tr>
<td></td>
<td>Altaic (1)</td>
<td>Turkish</td>
</tr>
<tr>
<td></td>
<td>Basque (1)</td>
<td>Basque</td>
</tr>
<tr>
<td></td>
<td>Dravidian (2)</td>
<td>Tamil, Telugu</td>
</tr>
<tr>
<td></td>
<td>Indo-European (21)</td>
<td>Ancient Greek, Bengali, Bulgarian, Catalan, Czech, Danish, Dutch,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>English, German, Greek, Hindi, Italian, Latin, Persian, Portuguese,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Romanian, Russian, Slovak, Slovenian, Spanish, Swedish</td>
</tr>
<tr>
<td></td>
<td>Japanese (1)</td>
<td>Japanese</td>
</tr>
<tr>
<td></td>
<td>Uralic (3)</td>
<td>Estonian, Finnish, Hungarian</td>
</tr>
<tr>
<td>Prague (7, 30)</td>
<td>Afro-Asiatic (1)</td>
<td>Arabic</td>
</tr>
<tr>
<td></td>
<td>Altaic (1)</td>
<td>Turkish</td>
</tr>
<tr>
<td></td>
<td>Basque (1)</td>
<td>Basque</td>
</tr>
<tr>
<td></td>
<td>Dravidian (2)</td>
<td>Tamil, Telugu</td>
</tr>
<tr>
<td></td>
<td>Indo-European (21)</td>
<td>Ancient Greek, Bengali, Bulgarian, Catalan, Czech, Danish, Dutch,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>English, German, Greek, Hindi, Italian, Latin, Persian, Portuguese,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Romanian, Russian, Slovak, Slovenian, Spanish, Swedish</td>
</tr>
<tr>
<td></td>
<td>Japanese (1)</td>
<td>Japanese</td>
</tr>
<tr>
<td></td>
<td>Uralic (3)</td>
<td>Estonian, Finnish, Hungarian</td>
</tr>
</tbody>
</table>
given by $n^{n-1}$, ranging from 9 possible trees for $n = 3$ to $10^9$ for $n = n^*$. From $n > n^*$ onwards, the number of trees grows too large to be manageable, so we resort to uniformly random sampling of $10^9$ trees for $n^* < n \leq 25$. For each tree in the collection, the classes it belongs to are indicated in the dataset.

The reason why we do not go beyond length 25 is that, for larger lengths, trees that belong to our classes under analysis are very scarce (Fig. 3A). For example, even sampling $10^9$ random trees for each length, no projective trees are found for $n > 18$. The same can be said of planar trees for $n > 19$, 1EC trees for $n > 22$, MH$_4$ trees for $n > 23$, and WG$_1$ trees for $n > 24$. For the MH$_5$ class, some trees can still be found in the sample for length 25, but only 69 out of $10^9$ belong to the class. Due to undersampling, the plot on artificial structures in the main article only shows points represented by at least 30 structures for $n > n^*$. 30 is considered a rule of thumb for the minimum sample size that is needed to estimate the mean of random variables that follow short tailed distributions (46). Fig. 3B shows average dependency distances not excluding any point.

For $n \leq n^*$, the ensemble of AS used to calculate $\langle d \rangle_{AS}$ contains all possible syntactic dependency structures for all classes. For $n > n^*$, it contains a random sample of them. Within a given ensemble, each structure is generated from a labelled directed tree whose vertex labels are interpreted as vertex positions in the linear arrangement. The values of $\langle d \rangle_{AS}$ for each class are exact (the mean over all possible syntactic dependency structures) for $n \leq n^*$ and random sampling estimates for $n > n^*$. A detailed explanation follows.

For a given $n$, an ensemble of syntactic dependency structures is generated with a procedure that is a generalization of the procedure used to generate random structures formed by an undirected tree and a linear arrangement (47). The procedure has two versions: the exhaustive version, that was used for $n \leq n^*$, and the random sampling version, that was used for $n > n^*$.

---

6The trees are freely available from [https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/XHRIYX](https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/XHRIYX)
Figure 3: Undersampling in artificial syntactic dependency structures (AS). A. $p$, the proportion of artificial structures of a certain class in the sample. B. The average dependency length, $\langle d \rangle_{AS}$, as a function of $n$, the sentence length (in words). For reference, the base line defined by a random linear arrangement of the words of the sentence, $\langle d \rangle_{rla}$ is also shown (dashed line).
The exhaustive version consists of

1. Generating all the $T(n)$ labelled (undirected) trees of $n$ vertices using Prüfer codes (48).
   It is known that $T(n) = n^{n-2}$ (49).

2. Converting each of these random trees into labelled directed trees (i.e., dependency trees) by rooting it in all possible ways. A rooting consists in choosing one node of the tree as the root, and making all edges point away from the root via a depth-first traversal. This produces $nT(n) = n^{n-1}$ syntactic dependency structures.

3. Producing a syntactic dependency structure from every directed tree using vertex labels (integers from 1 to $n$) as vertex positions in a linear arrangement (47).

4. Discarding the trees that do not belong to the target class.

The random sampling version consists of

1. Generating $S$ uniformly random labelled (undirected) trees of $n$ vertices, via uniformly random Prüfer codes (48).

2. Converting these uniformly random labelled trees to uniformly random labelled directed trees (i.e., dependency trees) by randomly choosing one node of each tree as the root, and making all edges point away from the root via a depth-first traversal. This produces $S$ syntactic dependency structures.

3. Same as exhaustive version.

4. Same as exhaustive version.

Note that Step 2 warrants that labelled directed trees in the ensemble are uniformly random: if we call $K_n$ the probability of generating each undirected tree of $n$ vertices with a random Prüfer
code, we can observe that each possible directed tree corresponds to exactly one undirected tree (obtained by ignoring arc directions), and each undirected tree corresponds to exactly \( n \) distinct directed trees (resulting from picking each of its \( n \) nodes as the root). Thus, the method of generating a random Prüfer code and then choosing a root generates each possible directed tree with a uniform probability \( K_n/n \) (as the probability of choosing the underlying undirected tree is \( K_n \), and the probability of choosing the relevant root is \( 1/n \)).

After each procedure, the average dependency length \( \langle d \rangle \) for a given \( n \) and a given class is calculated. While the exhaustive procedure allows one to calculate the true average dependency length over a certain class, the random sampling algorithm only allows one to estimate the true average. Put differently, the exhaustive procedure allows one to calculate exactly the expected dependency length in a class assuming that all labelled directed trees are equally likely whereas the random sampling procedure only allows one to obtain an approximation.

We explore all values of \( n \) within the interval \([n_{\text{min}}, n_{\text{max}}]\) with \( n_{\text{min}} = 3 \) and \( n_{\text{max}} = 25 \) and \( n^* = 10 \) and \( S = 10^9 \). The total number of syntactic dependency structures generated for our study is

\[
U = (n_{\text{max}} - n^*)S + \sum_{n=n_{\text{min}}}^{n^*} nT(n) = (n_{\text{max}} - n^*)S \sum_{n=n_{\text{min}}}^{n^*} n^{n-1}.
\]

Applying the parameters above, one obtains

\[
U \approx 1.6 \cdot 10^{10}
\]  
(2)

**Supplementary text**

**The random baseline**

Although the random baseline

\[
\langle d \rangle_{\text{rla}} = (n + 1)/3
\]  
(3)
follows from Jaynes’ maximum entropy principle in the absence of any constraint (50), it may be objected that our baseline is too unconstrained from a linguistic perspective. In previous research, random baselines that assume projectivity or consistent branching, whereby languages tend to grow parse trees either to the right (as in English) or to the left (as in Japanese), have been considered (26, 28, 51). However, it has been argued that these linguistic constraints could be a reflection of memory limitations (52, 53). Therefore, incorporating these linguistic constraints into the baseline for evaluating dependency distances would not provide an adequate test of the cognitive independence assumption because they could mask the effect of dependency distance minimization (DDm). Consistently, the planarity assumptions reduces the statistical power of a test of DDm (29). In addition, these additional constraints compromise the parsimony of a general theory of language for neglecting the predictive power of DDm (52).

_A priori_, $\langle d \rangle_{AS}$ could be below the random baseline as it occurs typically in human languages (25, 32) but it could also be above. As for the latter situation, empirical research in short sentences has shown that there are languages where dependency lengths are larger than expected by chance (29). In addition, there exist syntactic dependency structures where $\langle d \rangle > \langle d \rangle_{rla}$ from a network theoretical standpoint. For instance, among planar syntactic structures, the maximum average dependency distance is $\langle d \rangle_{max} = n/2$ (54).

$\langle d \rangle_{AS}$ never exceeds $\langle d \rangle_{rla}$ and it deviates from $\langle d \rangle_{rla}$ when $n = 3$ for projective trees, $n = 4$ for planar trees and $MH_4$ and $n = 5$ for $1EC$, $MH_5$ and $WG_1$. For the class of all syntactic dependency structures (Fig. 2 of main article), we find that $\langle d \rangle_{AS}$ matches Eq. [3] as expected from previous research (47).

**The classes of dependency structures**

_Planar trees:_ A dependency tree is said to be planar (or noncrossing) if its dependency arcs do not cross when drawn above the words. Planar trees have been used in syntactic parsing
Projective trees: A dependency tree is said to be projective if it is planar and its root is not covered by any dependency (see Figure 4). Projectivity facilitates the design of simple and efficient parsers (58, 59), whereas extending them to support non-projective trees increases their computational cost (60, 61). For this reason, and because treebanks of some languages (like English or Japanese) have traditionally had few or no non-projective analyses, many practical implementations of parsers assume projectivity (62, 63).

However, non-projective parsing is needed to deal with sentences exhibiting non-projective phenomena such as extraposition, scrambling or topicalization. Non-projectivity is particularly common in flexible word order languages, but generally present in a wide range of languages. However, non-projectivity in natural languages tends to be mild in the sense that the actually occurring non-projective trees are very close to projective trees, as they have much fewer crossing dependencies than would be expected by chance (64).

For this reason, there has been research interest in finding a restriction that would be a better fit for the phenomena observed in human languages. From a linguistic standpoint, the goal is to describe the syntax of human language better than with the overly restrictive projective trees or the arguably excessive permissiveness of admitting any tree without restriction, disregarding the observed scarcity of crossing dependencies. From an engineering standpoint, the goal is to
strike a balance between the efficiency provided by more restrictive parsers with a smaller search space and the coverage of the non-projective phenomena that can be found in attested sentences. In this line, various sets of dependency structures that have been proposed are supersets of projective trees allowing only a limited degree of non-projectivity. These sets are called mildly non-projective classes of dependency trees.

Here, we focus on three of the best known such sets, which have interesting formal properties and/or have been shown to be practical for parsing due to providing a good efficiency-coverage trade-off. We briefly outline them here, and refer the reader to (33) for detailed definitions and coverage statistics of these and other mildly non-projective classes of trees.

**Well-nested trees with Gap degree 1 (\(WG_1\)):** A dependency tree is well-nested (66) if it does not contain two nodes with disjoint, interleaving yields. Given two disjoint yields \(a_1 \ldots a_p\) and \(b_1 \ldots b_q\), they are said to interleave if there exist \(i, j, k, l\) such that \(a_i < b_j < a_k < b_l\). On the other hand, the gap degree of a tree is the maximum number of discontinuities present in the yield of a node, i.e., a dependency tree has gap degree 1 if every yield is either a contiguous substring, or the union of two contiguous substrings of the input sentence. Figure 5 provides graphical examples of these properties. \(WG_1\) trees have drawn interest mainly from the formal standpoint, for their connections to constituency grammar (67), but they also have been investigated in dependency parsing (68–70).
Figure 6: A. An 1-Endpoint-Crossing tree (given any dependency, dependencies crossing it are incident to a common node—for example, here the dependencies crossing the one marked in red are incident to node 4). B. A tree that is not 1-Endpoint-Crossing. The dependency arc in red has two crossing dependencies which are not incident to any common node.

Multi-Headed with at most $k$ heads per item ($MH_k$): Given $k \geq 3$, the set of $MH_k$ trees contains the trees that can be parsed by an algorithm called $MH_k$ (69). $k$ is a parameter of the class, such that for $k = 3$ the class coincides with projective trees, but for $k > 3$ it covers increasingly larger sets of non-projective structures (but the parser becomes slower). A recent neural implementation of the $MH_4$ parser has obtained competitive accuracy on UD (8). For $k > 4$, the $MH_k$ sets have been shown to be Pareto optimal (among known mildly non-projective classes) in terms of balance between efficiency and practical coverage (33). In this paper, we will consider the $MH_4$ and $MH_5$ sets.

1-Endpoint-Crossing trees (1EC): A dependency tree has the property of being 1-Endpoint-Crossing if, given a dependency, all other dependencies crossing it are incident to a common node (71). This property is illustrated in Figure 6. 1EC trees were the first mildly non-projective class of dependency trees to have a practical exact-inference parser (72), which was reimplemented with a neural architecture in (8). They are also in the Pareto frontier with respect to coverage and efficiency, according to (33).