Supplemental Materials for "Universals of word order result from optimization of grammars for efficient communication"

Michael Hahn
Department of Linguistics
Stanford University

Daniel Jurafsky
Department of Linguistics
Stanford University

Richard Futrell
Department of Language Science
University of California, Irvine

May 30, 2019

Contents

S1 Formalization of Correlations **2**

S2 Formalizing Comunicative Efficiency **3**
S2.1 Relation to Other Work . . . 3
S2.2 Possible values of λ . . . 4

S3 Supplementary Analyses for Study 1 **4**

S4 Supplementary Analyses for Study 2 **6**
S4.1 Correlation between Universals and Efficiency . . . 6
S4.2 Predictions for Individual Languages . . . 6
S4.3 Regression for Predicted Correlations . . . 6
S4.4 Comparing Efficiency to its Components . . . 9
S4.5 Results on all UD Relations . . . 9
S4.6 Other Experiments . . . 10

S5 Creating Optimized Grammars **10**
S5.1 Differentiable Ordering Grammars . . . 10
S5.2 Extracting Grammars from Datasets . . . 11
S5.3 Optimizing Grammars for Efficiency . . . 13

S6 Neural Network Architectures **15**

S7 Robustness to different language models and parsers **17**
S7.1 CKY Parsers . . . 17
S7.2 Distorted graph-based parsers . . . 18
S7.3 n-gram language models . . . 19

S8 Effects of data sparsity **19**

S9 Languages and Corpus Sizes **19**

S10 Dependency Length Minimization **21**
parseability vs parsability
where do we say that efficiency etc are reported on the held-out set?
notation U, L(T)
run czech parser
run controls on additional random grammars
word limit
grammar example in SI

cover letter goes to broad person
Czech parser
random all 50
keep in mind: pareto plot unify with SI plot

S1 Formalization of Correlations

Here we describe how we selected the correlations in Table 1 of the main paper, and how we formalized these using syntactic relations defined by Universal Dependencies.

We base our formalization on the comprehensive study by Dryer [1]. Greenberg's original study was based on 30 languages; more recently, Dryer [1] documented the word order correlations based on typological data from 625 languages. [1] formulated these universals as correlations between the order of objects and verbs and the orders of other syntactic relations. We test our ordering grammars for these correlations by testing whether the coefficients for these syntactic relations have the same sign as the coefficient of the verb-object relation. Testing correlations is therefore constrained by the degree to which these relations are annotated in UD. The verb-object relation corresponds to the *obj* relation defined by UD. While most of the other relations also correspond to UD relations, some are not annotated reliably. We were able formalize eleven out of Dryer's seventeen correlations in UD. Six of these could not be expressed individually in UD, and were collapsed into three coarse-grained correlations: First, tense/aspect and negative auxiliaries are together represented by the *aux* relation in UD. Second, the relation between complementizers and adverbial subordinators with their complement clauses is represented by the *mark* relation. Third, both the verb-PP relation and the relation between adjectives and their standard of comparison is captured by the *obl* relation.

The resulting operationalization is shown in Table S1. For each relation, we show the direction of the UD syntactic relation: $\rightarrow$ indicates that the verb patterner is the head; $\leftarrow$ indicates that the object patterner is the head.

Note that we follow [2] in converting the Universal Dependencies format to a format closer to standard syntactic theory, promoting adpositions, copulas, and complementizers to heads. As a consequence, the direction of the relations *case*, *cop*, and *mark* is reversed compared to Universal Dependencies. For clarity, we refer to these reversed relations as *lifted_case*, *lifted_cop*, and *lifted_mark*.

	Correlates with... verb	object	UD Relation	Greenberg [3]
①	adposition	NP	$\xrightarrow{lifted\_case}$	3, 4
②	copula verb	predicate	$\xrightarrow{lifted\_cop}$	–
③	tense/aspect auxiliary negative auxiliary	VP VP	$\xleftarrow{aux}$	16, 13 –
④	noun	genitive	$\xrightarrow{nmod}$	2, 23
⑤	noun	relative clause	$\xrightarrow{acl}$	24
⑥	complementizer adverbial subordinator	S S	$\xrightarrow{lifted\_mark}$	– –
⑦	adjective verb	std. of comp. PP	$\xrightarrow{obl}$	– 22
⑧	'want'	VP	$\xrightarrow{xcomp}$	15

Table S1: Greenbergian Correlations based on Dryer [1], with operationalizations with Universal Dependencies using the modified format of [2] (see text). For reference, we also provide the numbers of the closest corresponding universals stated in Greenberg's original study (if possible).

Excluded Correlations We excluded three correlations that are not annotated reliably in UD, and are only relevant to some of the world's languages: Question particles, plural words (i.e., independent plural markers), and articles. All three types of elements occur at most in parts of the 51 UD languages, and none of them is annotated reliably in those languages where they occur. Among these three types of elements, the one most prominent in our sample of 51 languages is articles (occurring in many European languages); UD subsumes them under the *det* relation, which is also used for other frequent elements, such as demonstratives and quantifiers.

We also excluded the verb-manner adverb correlation. UD does not distinguish manner adverbs from other elements labeled as adverbs, such as sentence-level adverbs and negation markers, whose ordering is very different from manner

adverbs. All types of adverbs are unified under the *advmod* relation. In the real orderings in our sample of 51 UD languages, the dominant ordering of *advmod* almost always matches that of subjects – that is, *advmod* dependents are ordered after the verb only in VSO languages. This ordering behavior is very different from that documented for manner adverbs by Dryer.

We further excluded the verb-subject correlation, which is not satisfied by much more than half of the world's languages (51 % among those with annotation in [4], with clear violation in 35.4 %). It is satisfied only in 33% of our sample of 51 UD languages, as quantified using the grammars we extracted. Dryer [1] counts this as a correlation since he describes the distribution of subject order as an interaction between a weak correlation with object order, and a very strong dominance principle favoring SV orderings. We focus on the modeling of correlations, and leave dominance principles to future research. We therefore excluded this correlation here.

Other Greenberg Universals Greenberg [3] stated a total of 45 universals. Twenty of these concern the structure of individual words (as opposed to word order, which we focus on here), and many of those have been argued to be explained by the "dual pressures" idea [**?**]. The other 25 universals concern word order; Dryer [1] reformulated most of these as correlations with verb-object order; these form the basis of our formalization in Table S1. There are a few other well-supported word order universals that are not correlations with verb-object order. This includes dominance principles [**?**] such as the strong preference for subjects to precede objects. Furthermore, there has been interest in Greenberg's universals 18 and 20, which describe correlations not with verb-object order, but of different elements of noun phrases [**?** **?** **?**]. Future work should examine whether these universals can also be linked to efficiency optimization.

S2 Formalizing Comunicative Efficiency

S2.1 Relation to Other Work

Here we discuss how our formalization of communicative efficiency relates to other information-theoretic work on language. Our formalization of efficiency is based on the function proposed in [5, 6, 7, 8] as a general efficiency metric for communicative systems. The key idea is to maximize the **amount of information** that linguistic forms provide about meanings, while constraining **complexity and diversity** of forms. Formally, if S denotes signals (e.g., words, sentences) and R denotes their referents (e.g., objects in a reference game), then this efficiency metric takes the form (notation slightly varies across these publications):

$$I[S, R] - \lambda H[S] \tag{1}$$

where $I[S, R]$ describes the **informativity** of the signals S about their referents R, and $H[S]$ describes the **complexity** of the communication system. While prior studies [5, 9, 10, 11] mostly considered settings where the signals S are individual words without further structure, the signals are entire sentences $\mathcal{U}$ in our setting. The underlying messages R which the speaker aims to convey are the syntactic structures $\mathcal{T}$. By the principle of compositionality [12], the meaning of a sentence is a function of the meanings of the parts and how they are combined. The syntactic structure specifies how the meanings of words are combined; therefore, recovering the syntactic structure is a prerequisite to understanding a sentence correctly.

We therefore arrive at the following efficiency metric:

$$R_{Eff} := R_{Parseability} + \lambda R_{Pred} \tag{2}$$

where **parseability** is the amount of information that utterances provide about their underlying syntactic structures:

$$R_{Pars} := I[\mathcal{U}, \mathcal{T}] = \sum_{t,u} p(t, u) \log \frac{p(t|u)}{p(t)} \tag{3}$$

and **predictability** is the negative entropy or surprisal of the language:

$$R_{Pred} := -H[\mathcal{U}] = \sum_{u} p(u) \log p(u) \tag{4}$$

This efficiency metric (1) is also equivalent to a deterministic version [13] of the Information Bottleneck approach, which has been successfully applied to modeling word meaning across different domains [11, 14]. The Information Bottleneck models complexity using a mutual information term instead of the entropy. This mutual information formalization is appropriate to codes that are nondeterministic [11]. As we assume deterministic grammars that transduce every underlying syntactic structure into one surface order, our model corresponds to the Deterministic Information Bottleneck [13], which arises

when the encoder is deterministic, and which uses the entropy as the complexity measure. Entropy also arises naturally as a complexity measure from the role played by surprisal in determining human language processing effort [15, 16, 17].

A few other approaches share the mutual information term for informativity, while using complexity measures that are not explicitly information-theoretic. In [18, 19, 20], the complexity function is the number of different forms; in [9] it is the difficulty of defining the concepts that are encoded. The number of forms is not applicable in our case, as languages will typically have infinitely many sentences. Notably, the models in [18, 20] have since been reformulated successfully in the Information Bottleneck formalism [11, 14], bringing them even closer to our formalization of efficiency.

In addition to these models, which quantify the efficiency of communication systems, there is closely related work formalizing the optimal choice of specific utterances in context, such as models of pragmatic reasoning in reference games [21, 22, 23]. In line with the other models, these assume that rational speakers choose utterances to optimize informativity about the referent object, and trade this off with the cost of the utterance, which is partly chosen to be the surprisal of the utterance [24].

S2.2 Possible values of λ

In the efficiency objective

$$R_{Eff} := R_{Pars} + \lambda R_{Pred} \tag{5}$$

the value of λ is constrained to be in $[0, 1)$. This means, surprisal must be weighted less strongly than parseability.

Greater values of λ can mathematically result in degenerate solutions. To show this, note that the following inequality always holds:

$$I[\mathcal{U}; \mathcal{T}] \leq H[\mathcal{U}]. \tag{6}$$

Therefore, if $\lambda \geq 1$, the efficiency objective satisfies

$$R_{Eff} = I[\mathcal{U}; \mathcal{T}] - \lambda H[\mathcal{U}] \leq 0. \tag{7}$$

and it takes the maximally possible value zero if there is only a single utterance $\mathcal{U}$, in which case both $I[\mathcal{U}; \mathcal{T}]$ and $H[\mathcal{U}]$ are zero. This is a degenerate language with only a single utterance, which is simultaneously used to convey all meanings. While the design of our word order grammars precludes a collapse of all trees to a single utterance, this shows that an objective with $\lambda \geq 1$ cannot be a generally applicable objective for language efficiency. In conclusion, λ is constrained to be in $[0, 1)$, with values closer to 1 placing similar weights on both predictability and parseability, whereas values closer to 0 diminish the role of predictability.

In our experiments, we chose $\lambda = 0.9$ as a mathematically valid value that puts similar weight on both predictability and parseability. The computational cost of grammar optimization precluded repeating the experiment for many values of λ. To tease apart the relative contributions of the two components, we also examine word order predictions for grammars optimized only to optimize parseability, or only predictability. As shown in Table S4, each of the eight correlations is predicted by at least parseability or predictability, without any contradictory predictions. That is, at λ close to its maximal value, the predictions of optimizing the two scoring functions individually add up to the predictions of efficiency optimization. Small values of λ correspond to the case where predictability plays no role, and only parseability is optimized (Table S4), in which case not all correlations are predicted (Figure S4). We also verified the robustness of our results to small variations of λ, by reporting results with $\lambda = 1$ in Table S6.

S3 Supplementary Analyses for Study 1

Per-Language Results In Figure S1, we show the predictability-parseability planes for every one the 51 languages, together with Pareto frontiers estimated from optimized grammars.

We z-transformed on the level of individual languages, normalizing the mean and SD parseability and predictability of the (1) real grammar, (2) the mean of predictability and parseability of all random grammars, (3) the grammar optimized for efficiency (at $\lambda = 0.9$, see Section S2.2), (4) grammar optimized for parseability only, and (5) grammar optimized for predictability only.

Within each language, we estimate the Pareto curve based on the points obtained by optimizing for (1) efficiency (at $\lambda = 0.9$, see Section S2.2), (2) parseability, and (3) predictability.

Figure (REF) in the main paper shows the average of these per-language plots.

Figure S1: Predictability and parseability of 51 languages. Green: random baselines, Red: real grammar, blue: approximate Pareto frontier, computed from the optimized grammars. All data are z-scored. Note that in a few languages, the real grammar is at a position slightly *beyond* the estimated Pareto frontier. This reflects noise in the optimization process due to use of stochastic gradient descent, i.e., the actual Pareto frontier might be somewhat further away from the baselines than the estimated one.

Robustness In the main paper, we tested whether real grammars are more efficient than the mean of random grammars, using a t-test. We also conducted the analysis using a Binomial test (one-sided), testing whether the real gramar is more efficient than the *median* of random grammars, avoiding any distributional assumption on the random grammars. As before, we used Hochberg's step-up procedure (Note that the tests for different languages are independent, as different random grammars are evaluated for each language), with $\alpha = 0.05$. In this analysis, real grammars improved in parseability for 80% of languages, in predictability for 69% of languages, and in either of both in 92% of languages ($p < 0.05$, with Bonferroni correction). In Table S2, we provide per-language results for the t-tests and binomial tests.

S4 Supplementary Analyses for Study 2

S4.1 Correlation between Universals and Efficiency

In Figure S2, we plot efficiency, parseability, and predictability (all are z-scored within language, as in Study 1) as a function of the number of satisfied correlations, for the real grammars of the 51 languages.

We found very similar results using Spearman's rank correlation (Efficiency: $\rho = 0.59$, $p = 9.8 \cdot 10^{-6}$; Parseability: $\rho = 0.55$, $p = 4.7 \cdot 10^{-5}$; Predictability: $\rho = 0.36$, $p = 0.012$).

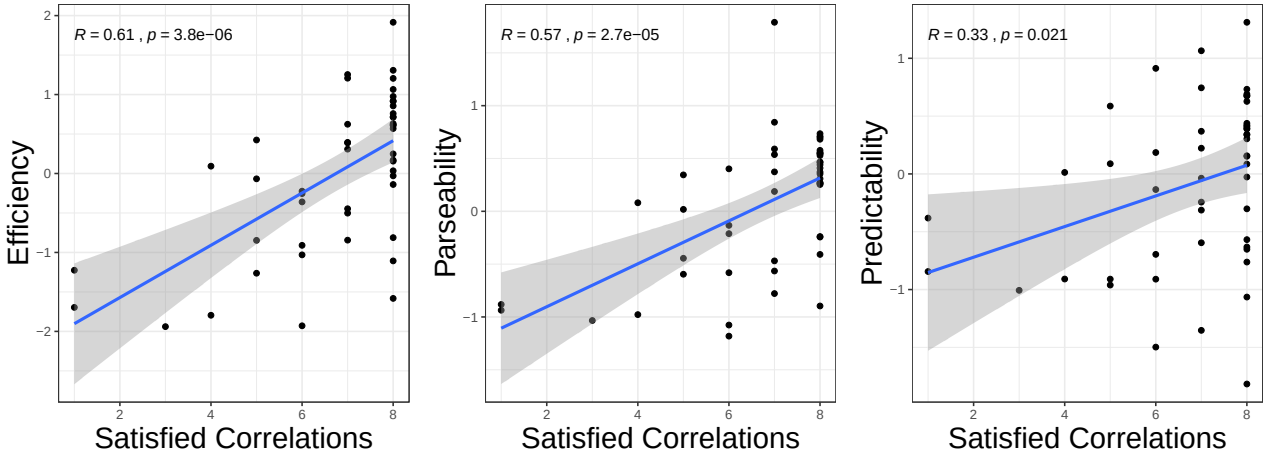

Figure S2: Correlation between the number of satisfied correlations (x-axis) and efficiency, parseability, and predictability (y-axis), for the 51 real languages.

S4.2 Predictions for Individual Languages

We show predictions for the eight correlations on the level of individual languages in Figure S3. We obtained these predictions for individual languages and each of the eight relations as follows. For each language and each of the objective functions (efficiency, predictability, parseability), we considered the optimized grammar that yielded the best value of this objective function among the eight optimized grammars (i.e., the grammar where the optimization procedure had been most successful). We interpreted this grammar as verb-object or object-verb depending on the order in the real grammar of the language.

S4.3 Regression for Predicted Correlations

Bayesian Regression We modeled the probabilities $p_{L,j}$ that a grammar optimized for data from language L satisfies the j-th correlation ($j = 1, ..., 8$) using a multilevel logistic model [25], with random intercepts for the language for whose data the grammar had been optimized, and for its language family, annotated according to `http://universaldependencies.org/`. Formally,

$$logit(p_{L,j}) = \alpha_j + u_{L,j} + v_{f_L,j} \tag{8}$$

where f_L is the language family of L. The intercepts α_j ($j = 1, ...8$) encode the population-level prevalence of the correlations when controlling for differences between datasets from different languages and language families; $u_{L,j}$, $v_{f_L,j}$ encode per-language and per-family deviations from the population-level intercept α_j.

Language	Pred. (t)	Parse. (t)	Pred. (Binomial)			Parseab. (Binomial)		
	p	p	Est.	CI	p	Est.	CI	p
Afrikaans	5.29×10^{-13}	1.46×10^{-6}	0.96	[0.89, 1]	1.59×10^{-13}	0.8	[0.69, 1]	7.01×10^{-6}
Ancient Greek	1.17×10^{-7}	0.998	0.8	[0.69, 1]	7.01×10^{-6}	0.33	[0.22, 1]	0.997
Arabic	0.0774	$<2 \times 10^{-16}$	0.57	[0.44, 1]	0.196	0.98	[0.92, 1]	1.55×10^{-15}
Basque	2.69×10^{-13}	1	0.89	[0.79, 1]	2.9×10^{-9}	0.31	[0.21, 1]	0.999
Belarusian	1	$<2 \times 10^{-16}$	0.14	[0.07, 1]	1	1	[0.95, 1]	$<2 \times 10^{-16}$
Bulgarian	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	1	[0.94, 1]	8.88×10^{-16}	1	[0.95, 1]	$<2 \times 10^{-16}$
Catalan	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$
Chinese	1.56×10^{-6}	0.0115	0.75	[0.64, 1]	0.000 117	0.7	[0.58, 1]	0.002 28
Coptic	0.001 75	$<2 \times 10^{-16}$	1	[0.94, 1]	1.78×10^{-15}	1	[0.95, 1]	$<2 \times 10^{-16}$
Croatian	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	1	[0.94, 1]	4.44×10^{-16}	1	[0.95, 1]	$<2 \times 10^{-16}$
Czech	0.438	0.5	0.46	[0.34, 1]	0.756	0	[0, 1]	1
Danish	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$
Dutch	1.41×10^{-11}	2.33×10^{-7}	0.87	[0.77, 1]	6.54×10^{-9}	0.76	[0.65, 1]	5.68×10^{-5}
English	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	1	[0.94, 1]	1.78×10^{-15}	1	[0.95, 1]	$<2 \times 10^{-16}$
Estonian	0.942	$<2 \times 10^{-16}$	0.27	[0.18, 1]	1	1	[0.95, 1]	$<2 \times 10^{-16}$
Finnish	8.85×10^{-6}	$<2 \times 10^{-16}$	0.7	[0.58, 1]	0.002 74	1	[0.95, 1]	$<2 \times 10^{-16}$
French	4.22×10^{-9}	$<2 \times 10^{-16}$	1	[0.94, 1]	8.88×10^{-16}	0.98	[0.91, 1]	6×10^{-15}
Galician	8.48×10^{-15}	$<2 \times 10^{-16}$	1	[0.94, 1]	1.78×10^{-15}	0.95	[0.87, 1]	4.07×10^{-13}
German	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	0.98	[0.91, 1]	1.18×10^{-14}	1	[0.95, 1]	$<2 \times 10^{-16}$
Gothic	9.98×10^{-16}	2.21×10^{-5}	0.98	[0.91, 1]	6×10^{-15}	0.74	[0.62, 1]	0.000 268
Greek	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$
Hebrew	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$
Hindi	$<2 \times 10^{-16}$	3.43×10^{-8}	1	[0.95, 1]	$<2 \times 10^{-16}$	0.78	[0.66, 1]	2.6×10^{-5}
Hungarian	0.127	$<2 \times 10^{-16}$	0.66	[0.54, 1]	0.0135	1	[0.95, 1]	$<2 \times 10^{-16}$
Indonesian	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$
Irish	0.982	$<2 \times 10^{-16}$	0.09	[0.04, 1]	1	1	[0.95, 1]	$<2 \times 10^{-16}$
Italian	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	1	[0.94, 1]	4.44×10^{-16}	1	[0.94, 1]	4.44×10^{-16}
Japanese	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$
Korean	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$	0.98	[0.92, 1]	1.55×10^{-15}
Latin	3.97×10^{-9}	3.51×10^{-11}	0.79	[0.67, 1]	1.79×10^{-5}	0.85	[0.75, 1]	6.92×10^{-8}
Latvian	1.14×10^{-6}	$<2 \times 10^{-16}$	0.76	[0.65, 1]	5.68×10^{-5}	1	[0.95, 1]	$<2 \times 10^{-16}$
Lithuanian	0.000 234	$<2 \times 10^{-16}$	0.62	[0.5, 1]	0.0492	0.98	[0.91, 1]	6×10^{-15}
Marathi	1	6.7×10^{-13}	0.18	[0.1, 1]	1	0.9	[0.81, 1]	6.42×10^{-10}
Norwegian	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	1	[0.94, 1]	1.42×10^{-14}	1	[0.94, 1]	2.22×10^{-16}
Old Church Slavonic	1	0.000 429	0.19	[0.1, 1]	1	0.73	[0.62, 1]	0.000 343
Persian	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	1	[0.94, 1]	2.22×10^{-16}	1	[0.95, 1]	$<2 \times 10^{-16}$
Polish	3.57×10^{-8}	$<2 \times 10^{-16}$	0.8	[0.69, 1]	4.35×10^{-6}	1	[0.95, 1]	$<2 \times 10^{-16}$
Portuguese	0.008 14	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$
Romanian	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$	1	[0.95, 1]	$<2 \times 10^{-16}$
Russian	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	1	[0.94, 1]	4.44×10^{-16}	1	[0.94, 1]	2.22×10^{-16}
Serbian	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	1	[0.94, 1]	2.22×10^{-16}	1	[0.95, 1]	$<2 \times 10^{-16}$
Slovak	6.14×10^{-6}	$<2 \times 10^{-16}$	0.67	[0.54, 1]	0.0129	1	[0.95, 1]	$<2 \times 10^{-16}$
Slovenian	1.79×10^{-5}	$<2 \times 10^{-16}$	0.8	[0.69, 1]	7.01×10^{-6}	1	[0.95, 1]	$<2 \times 10^{-16}$
Spanish	5.09×10^{-13}	$<2 \times 10^{-16}$	1	[0.94, 1]	8.88×10^{-16}	1	[0.95, 1]	$<2 \times 10^{-16}$
Swedish	$<2 \times 10^{-16}$	$<2 \times 10^{-16}$	0.98	[0.91, 1]	6×10^{-15}	1	[0.95, 1]	$<2 \times 10^{-16}$
Tamil	5.43×10^{-13}	1	1	[0.94, 1]	1.78×10^{-15}	0.26	[0.16, 1]	1
Telugu	8.2×10^{-7}	1	0.8	[0.69, 1]	7.01×10^{-6}	0.22	[0.13, 1]	1
Turkish	6.95×10^{-7}	7.49×10^{-15}	0.88	[0.78, 1]	1.62×10^{-8}	0.94	[0.86, 1]	2.76×10^{-12}
Ukrainian	5.79×10^{-15}	$<2 \times 10^{-16}$	0.87	[0.77, 1]	6.54×10^{-9}	1	[0.95, 1]	$<2 \times 10^{-16}$
Urdu	1	7.27×10^{-11}	0.1	[0.04, 1]	1	0.85	[0.74, 1]	2.02×10^{-7}
Vietnamese	0.002 74	$<2 \times 10^{-16}$	0.54	[0.41, 1]	0.333	1	[0.95, 1]	$<2 \times 10^{-16}$

Table S2: Per-language results in Study 1. For each language, we show the following: (1) p-values obtained from a one-sided t-test, for the null that the mean predictability/parseability of random grammars is at least as high as that of the real grammar. (2) Results from one-sided binomial tests, for the null that the the real grammar is better than at most 50% of random grammars. In addition to the p-value, we report point estimate and 95% confidence interval for the fraction of random grammars that have values below real grammars.

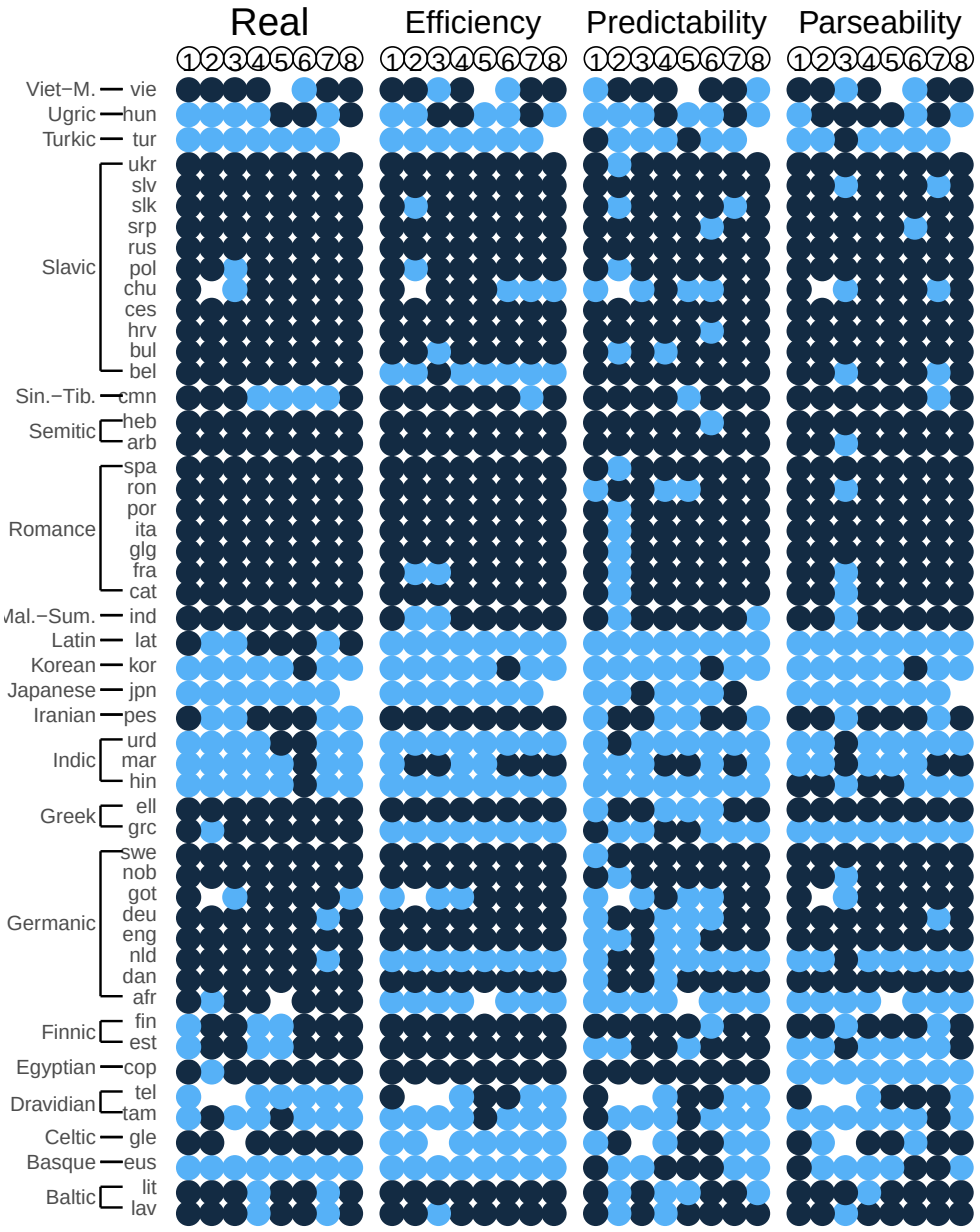

Figure S3: Order of the eight correlates across 51 languages, in the real grammars (left) and predicted by optimizing for efficiency, predictability, parseability (right). Dark blue: Verb patterner *precedes* object patterner (English, Arabic, ...). Light blue: Verb patterner *follows* object patterner (Japanese, Hindi , ...). White cells indicate that the relation is not annotated in the dataset for the given language.

	Prevalence	Bayesian			Frequentist			
		Mean	SD	$p(\beta \leq 0)$	β	SE	z	p
①	0.779	1.449	0.273	$< 1 \times 10^{-4}$	1.395	0.222	6.277	3.5×10^{-10}
②	0.678	0.761	0.171	1.0×10^{-4}	0.784	0.135	5.796	6.8×10^{-9}
③	0.696	1.003	0.424	0.012	0.943	0.342	2.753	0.006
④	0.782	1.586	0.318	$< 1 \times 10^{-4}$	1.512	0.251	6.013	1.8×10^{-9}
⑤	0.793	1.505	0.327	$< 1 \times 10^{-4}$	1.434	0.272	5.281	1.3×10^{-7}
⑥	0.757	1.133	0.43	0.006	1.072	0.352	3.041	0.002
⑦	0.748	1.093	0.388	0.003	1.026	0.322	3.185	0.001
⑧	0.911	3.854	0.878	$< 1 \times 10^{-4}$	3.823	0.782	4.887	1.0×10^{-6}

Table S3: Detailed results for Bayesian and Frequentist mixed-effects analyses for the eight correlations. We show (1) the raw prevalence of each correlation in the optimized grammars (8 grammars for each of the 51 languages), (2) for the Bayesian analysis, we provide posterior mean and SD of β, and the posterior probability that β has the opposite sign, (3) for the Frequentist analysis, we provide the point estimate, SE, z, and p-values (2-sided). The frequentist analysis confirms the results of the Bayesian analysis.

Following the recommendations of [26, 27], we used as a very weakly informative prior a Student's t prior with $\nu = 3$ degrees of freedom, mean 0, and scale $\sigma = 10$ (i.e., the PDF p is $\frac{1}{\sigma} p_3(x/\sigma)$, where p_3 is the PDF of the t-distribution with $\nu = 3$). We used this prior for $\alpha_j, \sigma_{L,j}, \tau_{L,j}$. Note that a correlation that holds in 90% of cases corresponds to an intercept $\alpha \approx 2.19$ in the logistic model, well within the main probability mass of the prior.

We modeled full covariance matrices of per-language and per-family random intercepts over all eight correlations. We placed an LKJ prior ($\eta = 1$) on these matrices, as described in [27]. We used MCMC sampling implemented in Stan [28, 29] using the R package `brms` [30]. We ran four chains, with 5000 samples each, of which the first 2500 were discarded as warmup samples. We confirmed convergence using $\hat{R}$ and visual inspection of chains [25].

We obtained the posterior density plots in Figures (REF main, S4) by applying the logistic transformation ($x \mapsto \frac{1}{1+\exp(-x)}$) to the posterior samples of α_j (8).

Robustness To ascertain the robustness of our results, we also conducted a frequentist analysis using `lme4` [31]. For each of the correlations, we conducted a logistic mixed-effects analysis predicting whether a grammar satisfies the correlation, with random effects of language and language family. The results are shown in Table S3 together with those of the Bayesian analysis. The frequentist analysis agrees with the Bayesian model; all eight correlations are predicted to hold in more than half of the optimized grammars ($p < 0.01$ each).

Note that the Bayesian analysis also estimates a posterior distribution of the number of satisfied correlations (see Figure S4), providing an elegant solution to the multiple-comparisons problem arising from analysing the eight correlation.

S4.4 Comparing Efficiency to its Components

In Figure S4, we plot the posterior distribution of the number of correlations predicted to hold in most optimized grammars, as obtained from the Bayesian regression. For each posterior sample, we say that the j-th correlation holds if the value of α_j in that posterior sample is positive. In the figure, we plot the fraction of posterior samples in which a given number of correlations is satisfied. In addition to grammars optimized for efficiency, we also report the result for grammars optimized for predictability and for parseability alone. Efficiency predicts all eight correlations with high posterior probability; predictability and parseability alone do not.

S4.5 Results on all UD Relations

In this section, we provide the predicted prevalence of correlations between the *obj* dependency and all UD dependency types, along with the expected prevalence according to typological studies. We also report results for grammars optimized for predictability and parseability individually.

We considered all UD syntactic relations occurring in at least two of the 51 languages. In Table S4, we present the data for the eight correlations discussed in the main paper, and for those other relations for which the typological literature provides data.[1] Additionally, in Table S5 we present data for the other UD relations, for which either no typological data

[1]The *aux* syntactic relation in UD has the auxiliary (verb-patterner) as its dependent, and has direction *opposite* to the auxiliary-verb relation ③. Therefore, this relation is *anti-correlated* with the verb-object relation, while ③ is *correlated*. For simplicity, we display this as a corelation in this table.

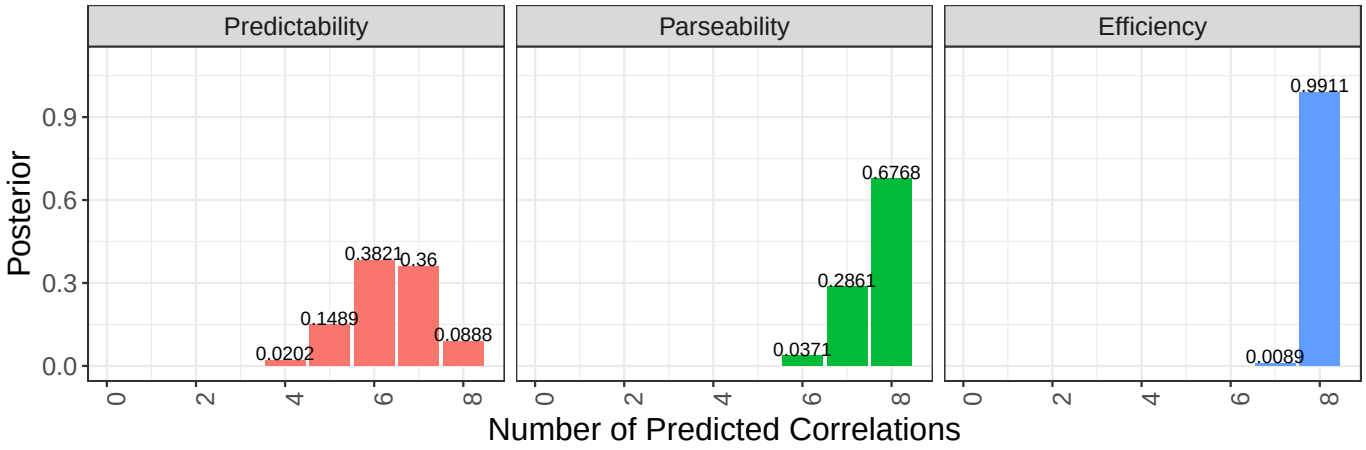

Figure S4: Posterior of the number of correlations correctly predicted by efficiency and its components, in the Bayesian multivariate mixed-effects logistic regression with random effects for languages and language families.

is available, or which are not linguistically meaningful.

Further, we report results for grammars optimized for dependency length minimization (DLM, see Section S10 for discussion). This has been suggested as a property of efficient word order in prior work, and, indeed, we found that DLM is predicted by efficiency optimization (Section S10). Results in Table S4 also show that optimizing for DLM makes predictions similar to efficiency optimization. See Section S10 for further discussion.

S4.6 Other Experiments

In Table S6 we report the results of our two previous, preregistered, simulations[2] together with results from the main experiment. These experiments all had the same setup described in Section S6, which was fixed before starting simulations; differences are that (1) one simulation places fully equal weight on parseability and predictability ($\lambda = 1.0$), and (2) the final experiment uses three random seeds per grammar. Results across all three experiments agree; jointly optimizing grammars for parseability and predictability produces all eight correlations.

S5 Creating Optimized Grammars

In this section, we describe the method we employ for creating grammars that are optimized for efficiency. We carry out optimization in an extended space of grammars that interpolates continuously between different grammars (Section S5.1). More specifically, we include probabilistic relaxations of grammars, which describe probability distributions over different ways of ordering a syntactic structure into a sentence. This makes efficiency a *differentiable* function of the grammar parameters, and enables efficient optimization with stochastic gradient descent, as we describe in Section S5.3.

This method addresses a major challenge noted in previous work optimizing grammars, namely that the predictability (and parseability) of an individual sentence depends on the entire distribution of the language. Previously, Gildea and Jaeger [35] optimized grammars for dependency length and trigram surprisal using a simple hill-climbing method on the grammar parameters, which required reestimating the trigram surprisal model in every iteration. Such a method would be computationally prohibitive for efficiency optimization, as it would require reestimating the neural network models after every change to the grammar, which would amount to reestimating them hundreds or thousands of times per grammar. Our method, by allowing for the use of stochastic gradient descent, addresses this challenge, as we describe in Section S5.3.

S5.1 Differentiable Ordering Grammars

We extended the parameter space of grammars by continuously interpolating between grammars, making efficiency a *differentiable* function of grammar parameters. The parameters of such a **differentiable word order grammar** are

[1]Note that we report an *anti-correlation*. The *aux* syntactic relation in UD has the auxiliary (verb-patterner) as its dependent, and has direction *opposite* to the auxiliary-verb relation ③. Therefore, this relation is *anti-correlated* with the verb-object relation, while ③ is *correlated*.

[2]`http://aspredicted.org/blind.php?x=8gp2bt`, `https://aspredicted.org/blind.php?x=bg35x7`. For the results of the locality simulations described in the first preregistration, see the DLM results in Table S4, see also Section S10 for discussion.

	Relation	Real	DLM	Pred	Pars	Efficiency	Expected Prevalence
①	lifted_case						> 50% [1]
②	lifted_cop						> 50% [1]
③	aux[1]						< 50% [1]
④	nmod						> 50% [1]
⑤	acl						> 50% [1]
⑥	lifted_mark						> 50% [1]
⑦	obl						> 50% [1]
⑧	xcomp						> 50% [1]
	advcl						> 50% [3, 70]
	ccomp						> 50% (cf. [71])
	csubj						> 50% (cf. [71])
	nsubj						See Section ??
	amod						≈ 50% [1]
	nummod						≈ 50% [72, 89A, 83A]

Table S4: Predictions on UD relations with predictions from the typological literature. The first section contains the eight correlations discussed in the main paper (See Section S1); the second section provides other relations for which predictions are available. In the last column, we indicate what direction would be expected typologically.

	English			Japanese		
Relation	Par.	a_τ	b_τ	Par.	a_τ	b_τ
object (*obj*)	0.1	0.04	−1.46	−0.1	0.99	−0.72
oblique (*obl*)	0.3	0.13	1.25	−0.3	0.99	0.73
case (*lifted_case*)	0.2	0.07	−0.89	−0.2	0.92	0.02

Figure S5: Sample Coefficients from grammars extracted from the real English and Japanese orderings, for the relations occurring in Figure 3 (Main Paper). We show parameters in $[-1, 1]$ for deterministic word order grammars, and the coefficients (a_τ, b_τ) for corresponding differentiable ordering grammars. For the deterministic grammars ('Par.'), positive coefficients indicate that the dependent will be placed after the head. For the differentiable grammars, $a_\tau > 0.5$ indicates ordering of dependents before heads, and larger b_τ indicates greater distance between head and dependent.

as follows. For each dependency label type τ, we have (1) a **Direction Parameter** $a_\tau \in [0, 1]$, and (2) a **Distance Parameter** $b_\tau \in \mathbb{R}$. Each dependent is ordered on the left of its head with probability a_τ and to the right with probability $1 - a_\tau$. Then for each set of co-dependents $\{s_1, \ldots, s_n\}$ placed on one side of a head, their order outward from the head is determined by iteratively sampling from the distribution $\text{softmax}(b_{\tau_1}, \ldots, b_{\tau_n})$ ([36], p. 184) without replacement.

If $a_\tau \in \{0, 1\}$, and the distances between values of b_τ (for different τ) become very large, such a differentiable grammar becomes deterministic, assigning almost full probability to exactly one ordering for each syntactic structure. In this case, the grammar can be converted into an equivalent grammar of the form described in Materials and Methods, by extracting a single parameter in $[-1, 1]$ for each relation τ.

S5.2 Extracting Grammars from Datasets

We extract grammars for the actual languages by fitting a differentiable ordering grammar maximizing the likelihood of the observed orderings. To prevent overfitting, we regularize each a_τ, b_τ with a simple Bayesian prior $logit(a_\tau) \sim \mathcal{N}(0, 1)$, $b_\tau \sim \mathcal{N}(0, 1)$. We then extract the posterior means for each parameter a_τ, b_τ, and convert the resulting differentiable grammar into an ordinary ordering grammar.

We provide an example in Figure S5, illustrating grammar parameters for the relations in Figure 3 of the main paper.

Relation	Real	DLM	Pred	Pars	Efficiency	Expected Prevalence
appos						Unknown
lifted_cc						Unknown
expl						Unknown
iobj						Unknown
vocative						Unknown
compound						Uninterpretable
det						Uninterpretable
dislocated						Uninterpretable
dep						Uninterpretable
advmod						Uninterpretable
conj						UD Artifact
discourse						UD Artifact
fixed						UD Artifact
flat						UD Artifact
goeswith						UD Artifact
list						UD Artifact
orphan						UD Artifact
parataxis						UD Artifact
reparandum						UD Artifact

Table S5: Predictions on UD relations for which no predictions are available in the typological literature. "Uninterpretable" UD relations are those which collapse so many different linguistic relationships that they are not linguistically meaningful. "UD artifact" relations are those whose order is determined strictly by UD parsing standards, such that their order is not linguistically meaningful: these include dependencies such as the connection between two parts of a word that have been separated by whitespace inserted as a typo (*goeswith*).

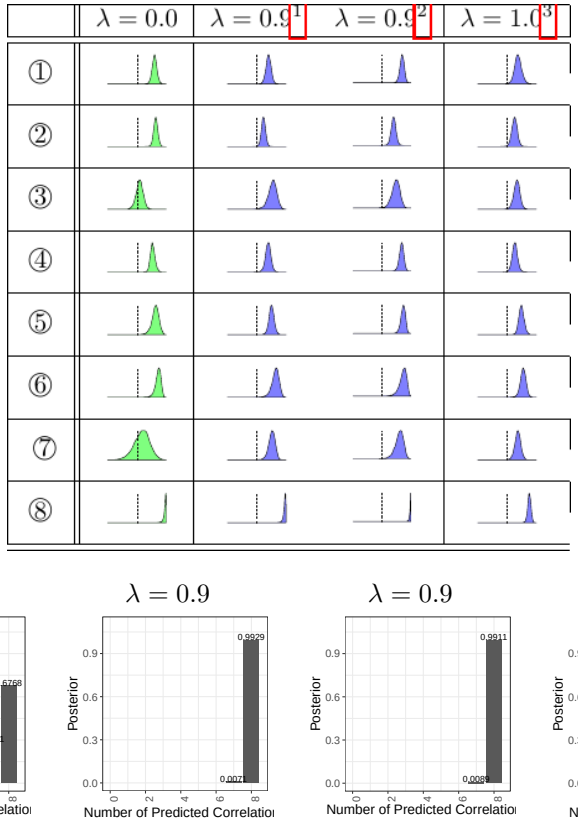

Table S6: Results from optimization experiments for different values of λ, including our two previous preregistered experiments. For comparison, we also show results for $\lambda = 0$, corresponding to optimizing for parseability only (same results as reported in Tables (S4-S5)). For $\lambda = 0.9$, we report results from one preliminary preregistered experiment (left) and the final experiment (right). For $\lambda = 1.0$, we report the other preliminary preregistered experiment. Giving similar weight to parseability and predictability – that is, λ close to 1 – results in more accurate word order predictions than choosing a small value of λ such as $\lambda = 0.0$. Note that λ cannot take values smaller than zero, or greater than one, see Section S2.2.

We validated the extracted grammars by comparing the dominant orders of six syntactic relations that are also annotated in the World Atlas of Linguistic Structures (WALS, [37]). Among the eight Greenbergian correlations that we were able to test, five are annotated in WALS. In Table S7, we compare our grammars with WALS on these five relations, and the verb-object relation. WALS has data for 74% of the entries[3], and lists a dominant order for 91% of these. The grammars we extracted from the corpora agree with WALS in 96 % of these cases.

S5.3 Optimizing Grammars for Efficiency

In this section, we describe how we optimized grammar parameters for efficiency. A word order grammar can be viewed as a function $\mathcal{L}_\theta$, whose behavior is specified by parameters θ, which takes an unordered dependency tree t as input and produces as output an ordered sequence of words $u = \mathcal{L}_\theta(t)$ linearizing the tree. More generally, if $\mathcal{L}_\theta$ is a differentiable ordering grammar (Section S5.1), then $\mathcal{L}_\theta(t)$ defines a *probability distribution* $p_{\mathcal{L}_\theta}(u|t)$ over ordered sequences of words u. In the limit where $\mathcal{L}_\theta$ becomes deterministic, the distribution $p_{\mathcal{L}_\theta}(u|t)$ concentrates on a single ordering u.

Recall the definition of efficiency

$$R_{Eff} := R_{Parseability} + \lambda R_{Pred} \tag{9}$$

where

$$R_{Pars} := I[\mathcal{U}, \mathcal{T}] = \sum_{t,u} p(t,u) \log \frac{p(t|u)}{p(t)} \tag{10}$$

[3]Serbian and Croatian are listed as a single language Serbian-Croatian in WALS. In the table, we compare those with the grammar we extracted for Croatian, noting that it fully agrees with the Serbian grammar.

Language	Objects		Adpositions		Compl.		Rel.Cl.		Genitive		PP	
Afrikaans	DH	?	HD	?	HD	?	–	?	HD	?	HD	?
Anc.Grk.	DH	?	HD	?	HD	?	HD	?	HD	?	HD	?
Arabic	HD	HD	HD	HD	HD	HD	HD	?	HD	HD	HD	HD
Basque	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH
Belarusian	HD	*	HD	?	HD	?	HD	HD	HD	HD	HD	*
Bulgarian	HD	HD	HD	HD	HD	HD	HD	HD	HD	*	HD	HD
Catalan	HD	HD	HD	HD	HD	?	HD	HD	HD	HD	HD	?
Chinese	HD	HD	HD	*	DH	?	DH	DH	DH	DH	DH	DH
Coptic	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD
Croatian	HD	HD	HD	HD	HD	HD	HD	?	HD	*	HD	?
Czech	HD	HD	HD	HD	HD	HD	HD	HD	HD	*	HD	?
Danish	HD	HD	HD	HD	HD	HD	HD	HD	*HD*	DH	HD	HD
Dutch	DH	*	HD	HD	HD	HD	HD	HD	HD	HD	DH	*
English	HD	HD	HD	HD	HD	HD	HD	HD	HD	*	HD	HD
Estonian	HD	HD	DH	DH	HD	HD	*DH*	HD	DH	DH	HD	HD
Finnish	HD	HD	DH	DH	HD	HD	*DH*	HD	DH	DH	HD	HD
French	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD
Galician	HD	?	HD	?	HD	?	HD	?	HD	?	HD	?
German	HD	*	HD	HD	HD	HD	HD	HD	HD	HD	DH	*
Gothic	HD	?	HD	?	HD	?	HD	?	HD	?	HD	?
Greek	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	?
Hebrew	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	?
Hindi	DH	DH	DH	DH	HD	HD	DH	*	DH	DH	DH	?
Hungarian	*DH*	HD	DH	DH	HD	HD	HD	*	DH	DH	DH	?
Indonesian	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD
Irish	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD
Italian	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	?
Japanese	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH
Korean	DH	DH	DH	DH	*HD*	DH	DH	DH	DH	DH	DH	?
Latin	DH	?	HD	?	HD	?	HD	?	HD	?	DH	?
Latvian	HD	HD	HD	HD	HD	HD	HD	HD	DH	DH	DH	?
Lithuanian	HD	HD	HD	HD	HD	HD	HD	HD	DH	DH	DH	?
Marathi	DH	DH	DH	DH	HD	*	DH	DH	DH	DH	DH	?
Norwegian	HD	HD	HD	HD	HD	HD	HD	HD	HD	*	HD	?
O.C.Slav.	HD	?	HD	?	HD	?	HD	?	HD	?	HD	?
Persian	DH	DH	HD	HD	HD	HD	HD	HD	HD	HD	DH	?
Polish	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	?
Portuguese	HD	HD	HD	HD	HD	?	HD	HD	HD	HD	HD	?
Romanian	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	?
Russian	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	?
Serbian	HD	?	HD	?	HD	?	HD	?	HD	?	HD	?
Slovak	HD	?	HD	?	HD	?	HD	?	HD	?	HD	?
Slovenian	HD	HD	HD	HD	HD	?	HD	?	HD	*	HD	?
Spanish	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD
Swedish	HD	HD	HD	HD	HD	HD	HD	HD	*HD*	DH	HD	HD
Tamil	DH	DH	DH	DH	DH	*	*HD*	DH	DH	DH	DH	DH
Telugu	DH	DH	DH	DH	DH	*	DH	DH	DH	DH	DH	?
Turkish	DH	DH	DH	DH	DH	*	DH	DH	DH	DH	DH	DH
Ukrainian	HD	HD	HD	HD	HD	HD	HD	HD	HD	?	HD	?
Urdu	DH	DH	DH	DH	HD	HD	HD	*	DH	DH	DH	?
Vietnamese	HD	HD	HD	HD	*DH*	HD	–	HD	HD	HD	HD	HD

Table S7: Comparing grammars extracted from databases to linguistic judgments in the World Atlas of Linguistic Structures. For each of the six syntactic relation, the first column provides the ordered coded in the extracted grammar; the second column provides the order coded in WALS (DH for dependent-head, HD for head-dependent order). '?' indicates that WALS has no data. $*$ indicates that WALS does not list a dominant order, which can mean that both orders occur at similar rates in the language, or that insufficient data was available when compiling WALS [38]. Finally, '–' indicates that the relation does not occur in the corpus.

$$R_{Pred} := -H[\mathcal{U}] = \sum_u p(u) \log p(u) \tag{11}$$

where $t \sim \mathcal{T}$ is a distribution over syntactic structures, and $u \sim p_{\mathcal{L}_\theta}(u|t)$ denotes the corresponding linearized sentences.

These quatities are estimated using two neural models, as described in Section S6: A **parser** recovers syntactic structures from utterances by computing a distribution $p_\phi(t|u)$, parameterized via parser parameters ϕ. The degree to which a parser with parameters ϕ succeeds in parsing a sentence u with structure t is[4]

$$R^\phi_{Pars}(u,t) = \log \frac{p_\phi(t|u)}{p(t)} \tag{12}$$

A **language model**, with some parameters ψ, calculates the word-by-word surprisal of an utterance:

$$R^\psi_{Pred}(u) = -\sum_{i=1}^{|u|} \log p_\psi(u_i|u_{1\ldots i-1}) \tag{13}$$

Using this, we can rewrite Efficiency (9), for a given grammar θ, equivalently into the parseability and predictability achieved with the best parser and language models:

$$R^\theta_{Eff} := \max_{\phi,\psi} R^{\theta,\phi,\psi}_{Eff} := \max_{\phi,\psi} \mathbb{E}_t \, \mathbb{E}_{u\sim p_\theta(u|t)} \left[R^\phi_{Pars}(u,t) + \lambda R^\psi_{Pred}(u) \right] \tag{14}$$

In order to find an optimal grammar θ, we thus need to solve

$$\arg\max_\theta \; \max_{\phi,\psi} R^{\theta,\phi,\psi}_{Eff} \tag{15}$$

Importantly, $R^{\theta,\phi,\psi}_{Eff}$ is differentiable in θ, ϕ, ψ:

$$\partial_\theta R^{\theta,\phi,\psi}_{Eff} = \mathbb{E}_t \, \mathbb{E}_{u\sim p_\theta(u|t)} \left[[\partial_\theta \log p_\theta(u|t)] \cdot \left(R^\phi_{Pars}(u,t) + \lambda R^\psi_{Pred}(u) \right) \right] \tag{16}$$

$$\partial_\phi R^{\theta,\phi,\psi}_{Eff} = \mathbb{E}_t \, \mathbb{E}_{u\sim p_\theta(u|t)} \left[\partial_\phi R^\phi_{Pars}(u,t) \right] \tag{17}$$

$$\partial_\psi R^{\theta,\phi,\psi}_{Eff} = \mathbb{E}_t \, \mathbb{E}_{u\sim p_\theta(u|t)} \left[\lambda \cdot \partial_\psi R^\psi_{Pred}(u) \right] \tag{18}$$

where 16 is derived using the *score-function* or *REINFORCE* trick [39]. Note that the the derivatives inside the expectations on the right hand sides can all be computed using backpropagation for our neural network architectures.

We can therefore apply stochastic gradient descent to jointly optimize θ, ϕ, ψ: In each optimization step, we sample a dependency tree t from the treebank, then sample an ordering from the current setting of θ to obtain a linearized sentence $\mathbf{w} \sim p_\theta(\cdot|t)$. Then we do a gradient descent step using the estimator given by the expressions in the square brackets in 16-18.

Optimizing for only parseability (or predictability) is very similar – in this case, the terms involving R^ϕ_{Pred} (or R^ψ_{Pars}) are removed.

At the beginning of the optimization procedure, we initialize all values $a_\tau := 0.5$, $b_\tau := 0$ (except for the *obj* dependency, for which we fix a_τ to 0 or 1, see Section S6). The neural parser and language model are also randomly initialized at the beginning of optimization. Empirically, we observe that optimizing differentiable ordering grammars for efficiency leads to convergence towards deterministic behavior, allowing us to extract equivalent deterministic grammars as described in Section S5.1.

See Section S6 for the stopping criterion and learning rates used in this optimization scheme.

S6 Neural Network Architectures

In this section, we describe the details of the neural network architectures. Choices follow standard practice in machine learning and were fixed before evaluating word order properties, and the efficiency of real languages.

[4]Note that, in the definition of R_{Pars}, the term $p(t)$ is a constant independent of ϕ and the word order grammar $\mathcal{L}_\theta$; it can therefore be ignored in the optimization process.

Optimization	Learning Rate	5e-6, 1e-5, 2e-5, 5e-5
	Momentum	0.8, 0.9
Language Model	Learning Rate	0.5, 0.1, 0.2
	Dropout Rate	0.0, 0.3, 0.5
	Embedding Size (Words)	50
	Embedding Size (POS)	20
	LSTM Layers	2
	LSTM Dimensions	128
Parser	Learning Rate	0.001
	Dropout Rate	0.2
	Embedding Size	100
	LSTM Layers	2
	LSTM Dimensions	200

Table S8: Hyperparameters

Estimating Predictability We choose a standard LSTM language model [41, 40]. We restrict the vocabulary to the most frequent 50,000 words in the treebanks for a given language. Given the moderate size of the corpora, this limit is only attained only for few languages. In each time step, the input is a concatenation of embeddings for the word, for language-specific POS tags, and for universal POS tags. The model predicts both the next word and its POS tags in each step. Using POS tags is intended to prevent overfitting on small corpora.

Estimating Parseability We use a biaffine attention parser architecture [42, 43, 44]. This architecture is remarkably simple: the words of a sentence are encoded into context-sensitive embeddings using bidirectional LSTMs, then a classifier is trained to predict the head for each work. The classifier works by calculating a score for every pair of word embeddings (w_i, w_j), indicating the likelihood that the jth word is the head of the ith word. This is a highly generic architecture for recovering graph structures from strings, and is a simplification of graph-based parsers which reduce the parsing problem to a minimal spanning tree problem [45].

To reduce overfitting on small corpora, we choose a delexicalized setup, parsing only from POS tags. Preliminary experiments showed that a parser incorporating word forms overfitted long before the ordering grammar had converged; parsing from POS tags prevents early overfitting. This decision was made before evaluating word order properties.

Hyperparameters Neural network models have hyperparameters such as the number of hidden units, and the learning rate. For predictability and parseability optimization, we first selected hyperparameters on the respective objectives for selected languages on the provided development partitions. These parameters are shown in Table S8. Then, for each language and each objective function, we created eight random combinations of these selected hyperparameter values, and selected the setting that yielded the best value of the respective objective function (efficiency, predictability, parseability) on the language. We then used this setting for creating optimized word order grammars.

All word and POS embeddings are randomly initialized with uniform values from $[-0.01, 0.01]$. We do not use pretrained embeddings [46]: While these could improve performance of language models and parsers, they would introduce confounds from the languages' actual word orders as found in the unlabeled data.

Improved Unbiased Gradient Estimator We employ two common variance reduction methods to improve the estimator (16), while keeping it unbiased. For predictability, note that the surprisal of a specific word only depends on the preceding words (not on the following words), and thus only depends on ordering decisions made up to that word. We represent the process of linearizing a tree as a dynamic stochastic computation graph, and use these independence properties to apply the method described in Schulman et al. [47] to obtain a version of (16) with lower variance. Second, we use a word-dependent moving average of recent per-word losses (the word's surprisal in the case of predictability, and the negative log-probability of the correct head and relation label in the case of parseability) as control variate [39]. These two methods reduce the variance of the estimator and thereby increase the speed of optimization and reduce training time, without biasing the results. For numerical stability, we represent $a_\tau \in [0, 1]$ via its logit $\in \mathbb{R}$. Furthermore, to encourage exploration of the parameter space, we add an entropy regularization term [48] for each Direction Parameter a_τ, which penalizes a_τ values near 0 or 1. The weight of the entropy regularization was chosen together with the other hyperparameters.[5]

[5] Explored values: 0.0001, 0.001.

These techniques for improving (16) are well-known in the machine learning literature, and we fixed these before evaluating optimized grammars for word order properties.

Learning Rate and Stopping Criterion We update word order grammar parameters θ using Stochastic Gradient Descent with momentum. For the language model parameters ϕ, we use plain Stochastic Gradient Descent without momentum, as recommended by Merity et al. [49]. For the parser parameters ψ, we use Adam [50], following Dozat et al. [44]. The learning rates and other optimization hyperparameters were determined together with the other hyperparameters.

All corpora have a predefined split in training and held-out (development) sets. We use the training set for optimizing parameters, and apply Early Stopping [51] using the held-out set.

For **estimating the parseability or predictability** of a given grammar, we optimize the neural model on data ordered according to this grammar, and report the parseability/predictability on the held-out set to avoid overfitting to the training set. For Early Stopping, we evaluate on the held-out set at the end of every epoch.

For **optimizing grammars**, we jointly apply gradient descent to the grammar parameters and the neural models, using the gradient estimator (16-18). For Early Stopping, we evaluate on the held-out set in intervals of 50,000 sentences, using a Monte-Carlo estimate of $R_{Eff}^{\theta,\phi,\psi}$ (S5.3), sampling a single linearized sentence for each syntactic structure in the held-out set. When reporting the parseability/predictability of an optimized grammar, we evaluate these values for its fully deterministic version (Section S5.1) to allow fair comparison with baseline grammars.

The choice of optimization methods and the stopping criterion were fixed before we investigated language efficiency or word order correlations.

Optimized Grammars As described in the main paper, for each language, we created 8 optimized languages for each optimization criterion. We enforced balanced distribution of object-verb and verb-object ordering among optimized languages by fixing a_τ for the *obj* dependency to be 0.0 in four of these languages, and 1.0 in the other four. This maximizes statistical precision in detecting and quantifying correlations between the *obj* relation and other relations.

For efficiency optimization, for each grammar, we ran efficiency optimization with three different random seeds, selecting among these the seed that yielded the best overall efficiency value. This helps control for variation across random seeds for the stochastic gradient descent optimization method.

S7 Robustness to different language models and parsers

Here we take up the question of the extent to which our results are dependent on the particular parser and language model used in the optimization process. We want to know: when we optimize a word order grammar for efficiency, have we produced a language which is highly efficient *in general*, or one which is highly efficient *for a specific parser*? We wish to argue that natural language syntax is optimized for efficiency in general, meaning that syntactic trees are highly recoverable from word orders in principle. If it turns out that our optimized languages are only optimal for a certain parser from the NLP literature, then we run the risk of circularity: it may be that the reason this parser was successful in the NLP literature was because it implicitly encoded word order universals in its inductive biases, and thus it would be no surprise that languages which are optimized for parseability also show those universals.

In this connection, we note that the parser and language model architectures we use are highly generic, and do not encode any obvious bias toward natural-language-like word orders. The LSTM language model is a generic model of sequence data which is also been used to model financial time series [52] and purely theoretical chaotic dynamical systems [53]; the neural graph-based parser is simply solving a minimal spanning tree problem [45]. Nevertheless, it may be the case that a bias toward word order universals is somehow encoded implicitly in the hyperparameters and architectures of these models.

Here we address this question by demonstrating that our languages optimized for efficiency are also optimal under a range of different language models and parsers. These results show that our optimization process creates language in which strings are generally predictable and informative about trees, without dependence on particular prediction and parsing algorithms.

S7.1 CKY Parsers

We constructed simple Probabilistic Context-Free Grammars (PCFGs) from corpora and word order grammars, using a simplified version of the models of [54] (Model 1). In our PCFGs, each head independently generates a set of left and right dependents. We formulate this as a PCFG where each rule has the form:

$$\mathrm{POS}_H \rightarrow \mathrm{POS}_H \ \mathrm{POS}_D$$

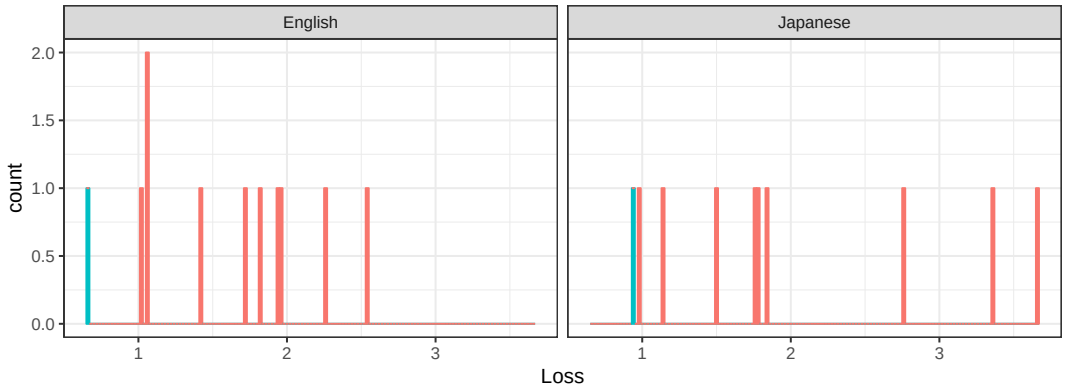

Figure S6: Parsing loss $H[\mathcal{T}|\mathcal{U}]$ (lower is better) computed by a simple CKY parser, for random word order grammars (red) and word order grammars optimized for efficiency (blue). We report $H[\mathcal{T}|\mathcal{U}]$ normalized by sentence length.

for head-initial structures, and

$$\text{POS}_H \rightarrow \text{POS}_D \ \text{POS}_H$$

for head-final structures, where each symbol is a POS tag. Thus, POS tags act both as terminals and as nonterminals.

We estimated probabilities by taking counts in the training partition, and performing Laplace smoothing with a pseudocount $\alpha = 1$ for each possible rule of this form. For such a PCFG, exact parsing is possible using Dynamic Programming, and specifically the CKY algorithm [55].

This parsing strategy is very different from the neural graph-based parser: While the graph-based parser solves a minimum spanning tree problem, the CKY algorithm uses dynamic programming to compute the exact probabilities of trees given a sentence, as specified by the generative model encoded in the PCFG. Second, while the graph-based neural parser uses machine learning to induce syntactic knowledge from data, the CKY parser performs exact probabilistic inference.

We used the CKY algorithm to compute the syntactic ambiguity $H[\mathcal{T}|\mathcal{U}]$ on the validation partition of the English and Japanese UD corpora, for random and optimized ordering grammars. Results (Figure S6) show that optimized grammars are more parseable for exact parsing of a simple PCFG.

S7.2 Distorted graph-based parsers

In this section, we address the idea that the graph-based parser might have a built-in bias toward certain kinds of orderings, and the question whether this might be responsible for our findings. In particular, we address the idea that the graph-based parser might have a bias toward parses involving short dependencies, which we call a **locality bias**. We address this by changing the order in which the parser sees words, in such a way that the distance between words in the input is not indicative of syntactic relations.

Even–odd order. A sequence of n words originally ordered as $w_1 w_2 w_3 w_4 \cdots w_n$ is reordered by separating the even and odd indices: $w_2 w_4 w_6 \cdots w_{n-1} w_1 w_3 w_5 \cdots w_n$ (assuming n odd). Therefore all words that are adjacent in the original order will be separated by a distance of $\approx n/2$ in the distorted order, while all words of distance 2 in the original order will become adjacent.

Interleaving order. In interleaving ordering, a sequence originally ordered as $w_1 w_2 w_3 \cdots w_n$ is split in half at the middle (index $m = \lceil n/2 \rceil$), and the two resulting sequences are interleaved, yielding $w_1 w_m w_2 w_{m+1} w_3 w_{m+3} \cdots w_n$. Thus all words that were originally adjacent will have distance 2 in the distorted order, with the intervening word coming from a very distant part of the sentence.

Inwards order. A sequence originally ordered as $w_1 w_2 w_3 \cdots w_{n-1} w_n$ is ordered from the edges of the string inwards, as $w_1 w_n w_2 w_{n-1} \cdots w_{\lceil n/2 \rceil}$. This corresponds to folding the string in on itself once, or equivalently, splitting the sequence in half at the middle, then interleaving the two resulting sequences after reversing the second one. The result is that the most non-local possible dependencies in the original order become the most local dependencies in the distorted order.

Sorted order. A sequence is reordered by sorting by POS tags, and randomizing the order within each block of identical POS tags. To each word, we then add a symbol encoding the original position in the sequence. For instance

PRON VERB PRON

may be reordered as

PRON 1 PRON 3 VERB 2

or

PRON 3 PRON 1 VERB 2

The numbers are provided to the parser as atomic symbols from a vocabulary ranging from 1 to 200; numbers greater than 200 (which may occur in extremely long sentences) are replaced by an out-of-range token.

The result is that distance between words in the input is not indicative at all of the presence of absence of syntactic relations between them.

Experiments Using English and Japanese data, we trained parsers for the ten random word order grammars and for the best grammar optimized for efficiency, with the input presented in each of the distorted orderings. Resulting parsing accuracy scores are shown in Figure S7. In all settings, the language optimized for efficiency achieved higher parsing accuracy than random ordering grammars, showing that the parser's preference for optimized languages cannot be attributed to a locality bias.

S7.3 n-gram language models

We model predictability using LSTM language models, which are are the strongest known predictors of the surprisal effect on human processing effort [56, 57]. In previous work, such as [35], predictability has often been measured using n-gram models.

Here, we show that languages optimized for LSTM predictability are also optimal for n-gram predictability. Specifically, we constructed bigram models with Kneser-Ney smoothing [58, 59]. A bigram model predicts each word taking only the previous word into account. This contrasts with LSTMs, which take the entire context into consideration. Thus, bigram models and LSTMs stand on opposing ends of a spectrum of language models taking more and more aspects of the context into account.

We estimated language models on the training partitions, and used the validation partitions to estimate surprisal. We conducted this for the ten random and the best optimized ordering grammars on English and Japanese data. Results (Figure S8) show that languages optimized for efficiency are optimal for a bigram language model.

S8 Effects of data sparsity

Here, we investigate whether the difference between real and baseline grammars is affected by the size of available datasets. If the difference between random and real grammars is due to data sparsity, we expect that it decreases as the amount of training data is increased. If, on the other hand, there is an inherent difference in efficiency between random and real grammars, we expect that the difference persists as training data is increased.

We considered Czech, the UD language with the largest amount of available treebank data (approx. 2.2 Million words), up to $\approx$ 300 times more data than is available for some other UD languages. We cosidered both a random ordering grammar, and the best ordering grammar optimized for parseabaility. For both of these ordering grammars, we trained the parser on successively larger portions of the training data (0.1 %, 1 %, 5%, 10%, 20 %, ..., 90 %, 100 %) and recorded parsing accuracy. Furthermore, for the random grammar, we varied the number of neurons in the BiLSTM (200, 400, 800) to test whether results depend on the capacity of the network.

The resulting curves are shown in Figure S9. A gap in parsing loss of about 0.2 nats appears already at 0.01 % of the training data (2000 words), and persists for larger amounts of training data.

S9 Languages and Corpus Sizes

In Table S9, we list the 51 languages with ISO codes and families, with the size of the available data per language. We included all UD 2.1 languages for which a training partition was available.

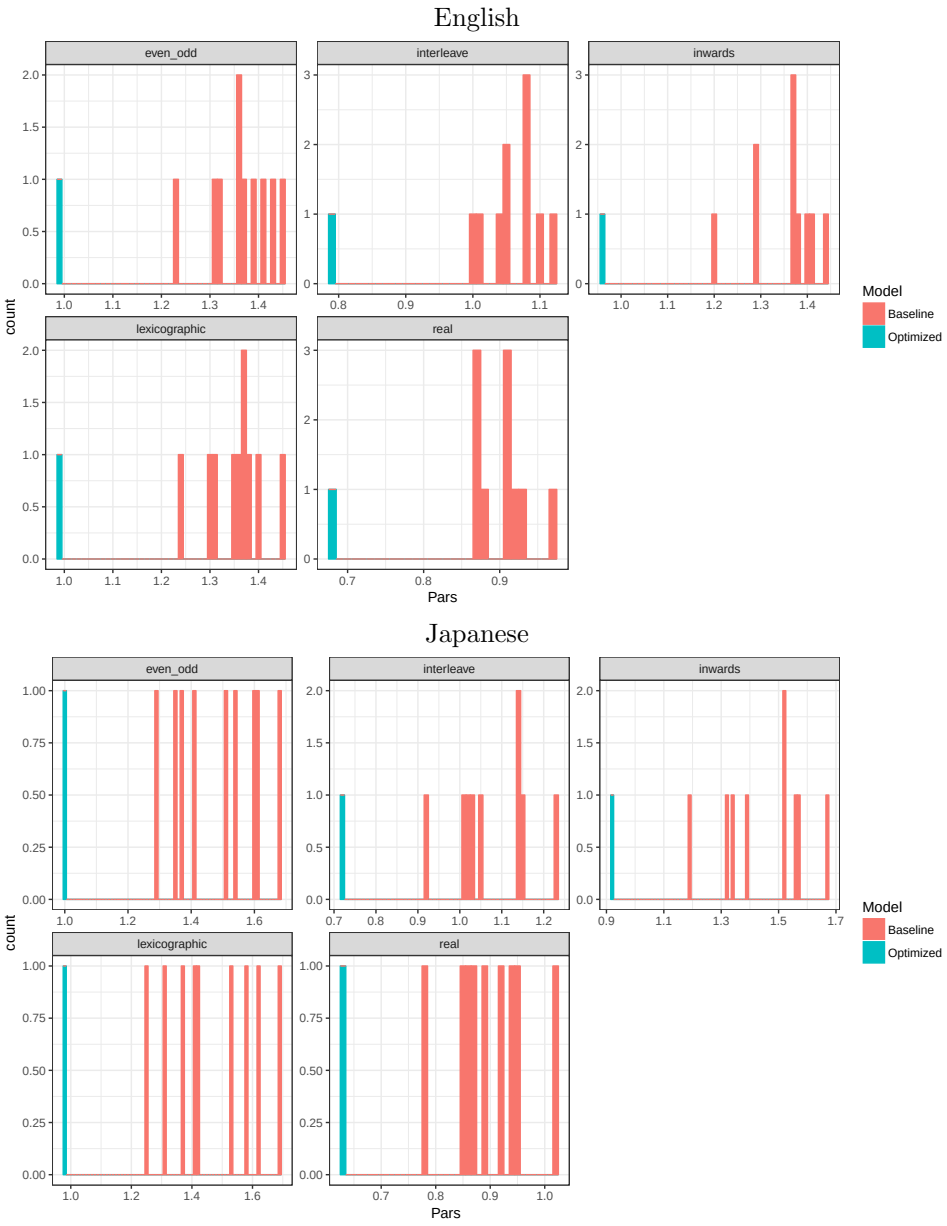

Figure S7: Parseability of baseline grammars and grammars optimized for efficiency, in English (top) and Japanese (bottom), measured by parsing loss $H[\mathcal{T}|\mathcal{U}]$ (lower is better), for the four distorted orderings, and the actual orderings ('real'). We report $H[\mathcal{T}|\mathcal{U}]$ normalized by sentence length.

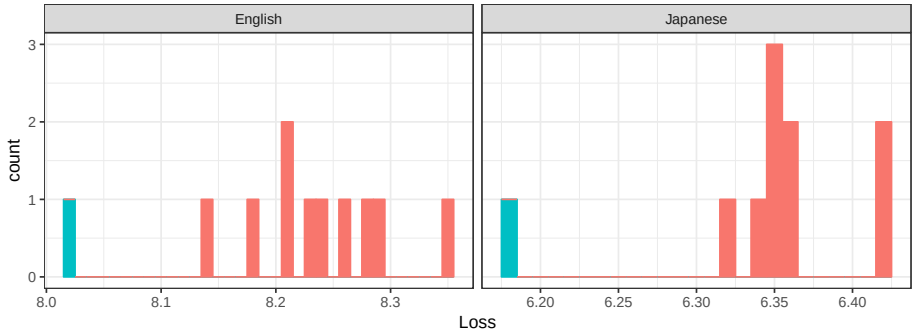

Figure S8: Surprisal (lower is better) computed from Bigram model, on English and Japanese data ordered according to random ordering grammars (red) and ordering grammars optimized for efficiency (blue).

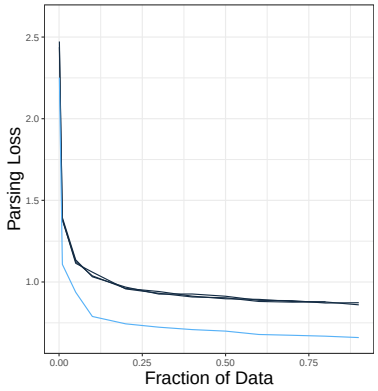

Figure S9: Parsing loss ($H[\mathcal{T}|\mathcal{U}]$, normalized by sentence length) for optimized (light blue) and random (black) ordering grammar on Czech data, as a function of the fraction of total training data provided.

S10 Dependency Length Minimization

Prior work has suggested *Dependency Length Minimization* (DLM) as a characteristic of efficient word order [2, 60, 61]. This is the idea that word order minimizes the average distance between syntactically related words. It is known that human languages reduce dependency length compared to random baselines [2, 60, 61].

Prior work has suggested principles akin to DLM as approximating efficiency optimization of grammars [62, 7, 63]. It is a heuristic formalization of the idea that long dependencies should create high memory requirements in parsing and prediction [62, 64, 65, 7]. Indeed, [7] argues specifically that it emerges from efficiency optimization.

Dependency length is typically quantified as the average distance between all pairs of syntactically related words, measured by the number of intervening words [2]. Dependency length quantified in this manner is a heuristic measure of complexity: The actual processing complexity induced by long dependencies is not a linear function of length and depends crucially on the types of dependencies involved [66] and the specific elements intervening between the head and dependent [64, 65, 67].

We asked whether efficiency optimization predicts dependency length minimization effects. We first computed dependency length for grammars optimized for efficiency. We found that 100% of grammars optimized for efficiency reduce average dependency length compared to baseline grammars ($p < 0.05$, by one-sided t-test). TODO This suggests that the reduction of dependency length is predicted by efficiency maximization. Next, we constructed grammars that minimize average dependency length, using the same gradient descent method as we used for efficiency optimization (Section S5.3). We expect that such grammars should have shorter dependency length than the real grammars, or grammars optimized for efficiency. In Figure S10, we plot the mean dependency length for optimized, real, and baseline orderings. We find that optimizing grammars for efficiency reduces dependency length to a similar degree as found in the actual orderings in the corpora, almost up to the limit given by directly optimizing for dependency length. We also plot more detailed results for four languages in Figure S11, plotting dependency length as a function of sentence length as reported in prior work [2]. Optiziming grammars for efficiency produces dependency lengths similar to those found in the actual orderings.

Next, we examined the word order properties of grammars optimized for DLM. In Table S4, we report the posterior

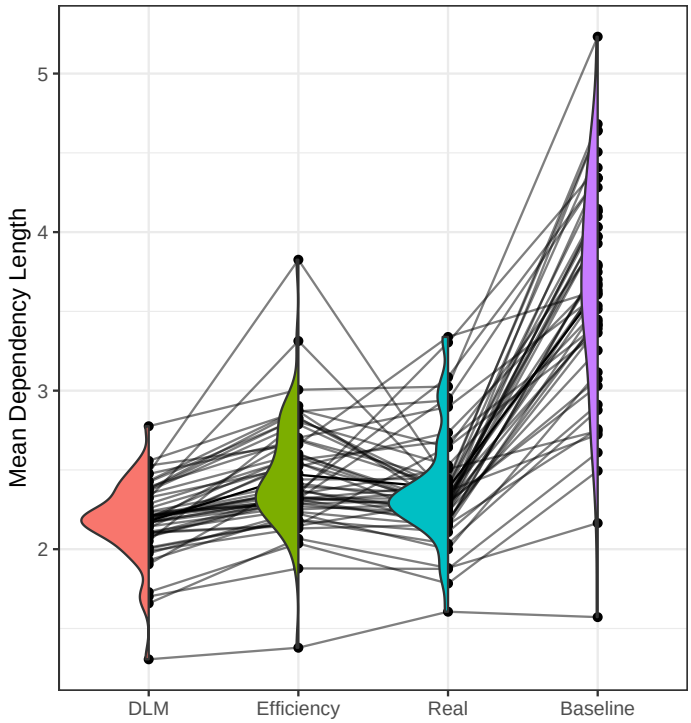

Figure S10: Average dependency length for grammars optimized to minimize dependency length (DLM, left), optimized for efficiency (second), the real orderings found in corpora (third), and random baseline grammars (right).

prevalence of word order correlations in grammars optimized for DLM show that optimizing for DLM makes predictions similar to efficiency optimization, itself a novel result. We find that these grammars also exhibit the eight correlations, similar to grammars directly optimized for efficiency. On other correlations, predictions of DLM also resemble those of efficiency optimization. However, it predicts strong correlations with *amod* (adjectival modifiers) and *nummod* (numeral modifiers) (see bottom of Table S4), which are not borne out typologically. In these cases, efficiency optimization predicts prevalnces closer to 50%, in line with typological data.

In conclusion, these results suggest that dependency length minimization is a by-product of efficiency optimization, providing support to theoretical arguments from the linguistic literature [62, 7, 63]. Furthermore, optimizing for dependency length correctly predicts a range of word order facts, though it appears to *overpredict* correlations when compared to direct optimization for efficiency.

References

[1] Matthew S Dryer. The Greenbergian word order correlations. *Language*, 68(1):81–138, 1992.

[2] Richard Futrell, Kyle Mahowald, and Edward Gibson. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341, 2015. doi: 10.1073/pnas.1502134112. URL `http://www.pnas.org/content/early/2015/07/28/1502134112.abstract`.

[3] Joseph H. Greenberg. Some universals of grammar with particular reference to the order of meaningful elements. In Joseph H. Greenberg, editor, *Universals of Language*, pages 73–113. MIT Press, Cambridge, MA, 1963.

[4] Matthew S. Dryer. Order of subject, object and verb. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig, 2013. URL `https://wals.info/chapter/81`.

[5] Ramon Ferrer i Cancho and Ricard V Solé. Least effort and the origins of scaling in human language. *Proceedings of the National Academy of Sciences*, 100(3):788–791, 2003.

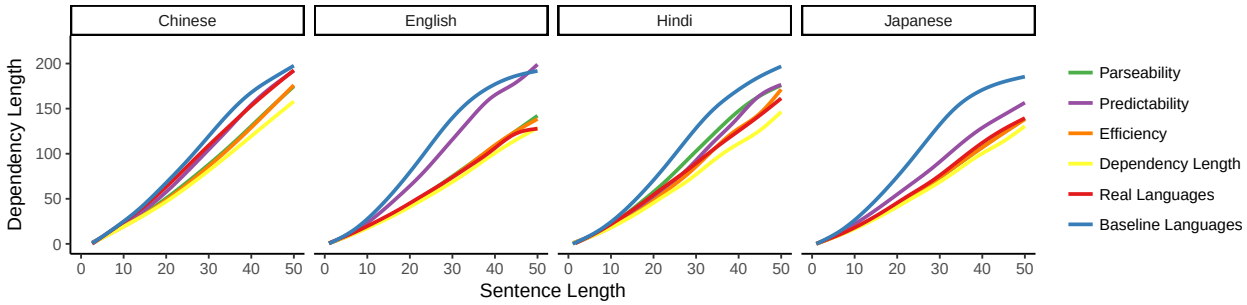

Figure S11: Total dependency length as a function of sentence length, for four diverse languages. We show results for optimized grammars (parseability, predictability, efficiency), for grammars specifically optimized to minimize dependency length, of the actual real orderings, and of the baseline grammars.

[6] Ramon Ferrer i Cancho and Albert Díaz-Guilera. The global minima of the communicative energy of natural communication systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(06):P06009, 2007.

[7] Richard Futrell. *Memory and locality in natural language*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2017.

[8] Benjamin Peloquin, Noah Goodman, and Mike Frank. The interactions of rational, pragmatic agents lead to efficient language structure and use. In *CogSci*, 2019.

[9] Charles Kemp and Terry Regier. Kinship categories across languages reflect general communicative principles. *Science*, 336(6084):1049–1054, 2012.

[10] Terry Regier, Charles Kemp, and Paul Kay. Word meanings across languages support efficient communication. In *The Handbook of Language Emergence*, pages 237–263. Wiley-Blackwell, Hoboken, NJ, 2015.

[11] Noga Zaslavsky, Charles Kemp, Terry Regier, and Naftali Tishby. Efficient compression in color naming and its evolution. *Proceedings of the National Academy of Sciences*, 115(31):7937–7942, 2018.

[12] Gottlob Frege. Über sinn und bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100(1):25–50, 1892.

[13] DJ Strouse and David J Schwab. The deterministic information bottleneck. *Neural computation*, 29(6):1611–1630, 2017.

[14] Noga Zaslavsky, Terry Regier, Naftali Tishby, and Charles Kemp. Semantic categories of artifacts and animals reflect efficient coding. In *CogSci*, 2019.

[15] John T. Hale. A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics and Language Technologies*, pages 1–8, 2001.

[16] Roger Levy. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177, 2008.

[17] Nathaniel J. Smith and Roger Levy. The effect of word predictability on reading time is logarithmic. *Cognition*, 128 (3):302–319, 2013.

[18] Terry Regier, Paul Kay, and Naveen Khetarpal. Color naming reflects optimal partitions of color space. *Proceedings of the National Academy of Sciences*, 104:1436–1441, 2007.

[19] Yang Xu and Terry Regier. Numeral systems across languages support efficient communication: From approximate numerosity to recursion. In P. Bello, M. Guarini, M. McShane, and B. Scassellati, editors, *Proceedings of the 36th annual meeting of the Cognitive Science Society*, pages 1802–1807, Austin, TX, 2014. Cognitive Science Society.

[20] Yang Xu, Terry Regier, and Barbara C. Malt. Historical semantic chaining and efficient communication: The case of container names. *Cognitive Science*, 40:2081–2094, 2016.

[21] Michael C Frank and Noah D Goodman. Predicting pragmatic reasoning in language games. *Science*, 336(6084): 998–998, 2012.

[22] Noah D. Goodman and Andreas Stuhlmüller. Knowledge and implicature: Modeling language understanding as social cognition. *Topics in Cognitive Science*, 5(1):173–184, 2013. doi: 10.1111/tops.12007.

[23] Justine T. Kao, Jean Y. Wu, Leon Bergen, and Noah D. Goodman. Nonliteral understanding of number words. *Proceedings of the National Academy of Sciences*, 111(33):12002–12007, 2014. doi: 10.1073/pnas.1407479111.

[24] Erin D Bennett and Noah D Goodman. Extremely costly intensifiers are stronger than quite costly ones. *Cognition*, 178:147–161, 2018.

[25] Andrew Gelman, Hal S Stern, John B Carlin, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

[26] Joyee Ghosh, Yingbo Li, Robin Mitra, et al. On the use of cauchy prior distributions for bayesian logistic regression. *Bayesian Analysis*, 13(2):359–383, 2018.

[27] Paul-Christian Bürkner. Advanced bayesian multilevel modeling with the r package brms. *The R Journal*, 10(1): 395–411, 2018.

[28] Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

[29] Matthew D Hoffman and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[30] Paul-Christian Bürkner. brms: An r package for bayesian multilevel models using stan. *Journal of Statistical Software, Articles*, 80(1):1–28, 2017.

[31] Douglas M. Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015. doi: 10.18637/jss.v067.i01.

[32] Holger Diessel. The ordering distribution of main and adverbial clauses: A typological study. *Language*, pages 433–455, 2001.

[33] Matthew Synge Dryer. The positional tendencies of sentential noun phrases in universal grammar. *Canadian Journal of Linguistics/Revue canadienne de linguistique*, 25(2):123–196, 1980.

[34] Matthew S. Dryer and Martin Haspelmath. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig, 2013. URL `https://wals.info/`.

[35] Daniel Gildea and T. Florian Jaeger. Human languages order information efficiently. *arXiv*, 1510.02823, 2015. URL `http://arxiv.org/abs/1510.02823`.

[36] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[37] M. Haspelmath, M.S. Dryer, D. Gil, and B.. Comrie. The World Atlas of Language Structures Online. 2005.

[38] Matthew S Dryer. The evidence for word order correlations. *Linguistic Typology*, 15(2):335–380, 2011.

[39] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[40] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[41] Yoav Goldberg. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309, 2017.

[42] Eliyahu Kiperwasser and Yoav Goldberg. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association of Computational Linguistics*, 4:313–327, 2016.

[43] Xingxing Zhang, Jianpeng Cheng, and Mirella Lapata. Dependency parsing as head selection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 665–676, Valencia, Spain, 2017.

[44] Timothy Dozat, Peng Qi, and Christopher D Manning. Stanford's graph-based neural dependency parser at the conll 2017 shared task. *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, 2017.

[45] Ryan T McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530. Association for Computational Linguistics, 2005.

[46] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[47] John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, pages 3528–3536, 2015.

[48] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[49] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing LSTM language models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. URL `https://openreview.net/forum?id=SyyGPP0TZ`.

[50] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference Learning Representations*, 2014.

[51] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.

[52] Justin Sirignano and Rama Cont. Universal features of price formation in financial markets: perspectives from deep learning. *arXiv preprint arXiv:1803.06917*, 2018.

[53] Olalekan Ogunmolu, Xuejun Gu, Steve Jiang, and Nicholas Gans. Nonlinear systems identification using deep dynamic neural networks. *arXiv preprint arXiv:1610.01439*, 2016.

[54] Michael Collins. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4): 589–637, 2003.

[55] Tadao Kasami. An efficient recognition and syntax-analysis algorithm for context-free languages. *Coordinated Science Laboratory Report no. R-257*, 1966.

[56] Stefan L. Frank and Rens Bod. Insensitivity of the human sentence-processing system to hierarchical structure. *Psychological Science*, 22(6):829–834, 2011.

[57] Adam Goodkind and Klinton Bicknell. Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018)*, pages 10–18, Salt Lake City, UT, 2018. Association for Computational Linguistics.

[58] Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184. IEEE, 1995.

[59] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393, 1999.

[60] Haitao Liu, Chunshan Xu, and Junying Liang. Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of Life Reviews*, 2017.

[61] David Temperley and Dan Gildea. Minimizing syntactic dependency lengths: Typological/cognitive universal? *Annual Review of Linguistics*, 4:1–15, 2018.

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

[63] Richard Futrell, Roger Levy, and Edward Gibson. Generalizing dependency distance: Comment on "dependency distance: A new perspective on syntactic patterns in natural languages" by haitao liu et al. *Physics of Life Reviews*, 21:197–199, 2017.

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

[65] E. Gibson. The dependency locality theory: A distance-based theory of linguistic complexity. In Alec Marantz, Yasushi Miyashita, and Wayne O'Neil, editors, *Image, Language, Brain: Papers from the First Mind Articulation Project Symposium*, pages 95–126, 2000.

[66] Vera Demberg and Frank Keller. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2):193–210, 2008. ISSN 0010-0277. doi: DOI:10.1016/j.cognition.2008.07.008.

[67] Richard L. Lewis and Shravan Vasishth. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, 29(3):375–419, 2005.

Language	ISO Code	Family	Sentences (train/held-out)	Words (train/held-out)
Afrikaans	afr	Germanic	1315/194	30765/4808
Ancient Greek	grc	Greek	26322/2156	323993/33468
Arabic	arb	Semitic	21864/2895	737410/93666
Basque	eus	Basque	5396/1798	61040/20122
Belarusian	bel	Slavic	260/65	4328/1274
Bulgarian	bul	Slavic	8907/1115	106813/13822
Catalan	cat	Romance	13123/1709	375524/50954
Chinese	cmn	Sino-Tibetan	3997/500	85013/10899
Coptic	cop	Egyptian	364/41	8818/871
Croatian	hrv	Slavic	7689/600	148560/12922
Czech	ces	Slavic	102993/11311	1547431/163578
Danish	dan	Germanic	4383/564	69273/8952
Dutch	nld	Germanic	18310/1518	234859/19115
English	eng	Germanic	17062/3070	263328/39537
Estonian	est	Finnic	6959/855	69754/8709
Finnish	fin	Finnic	27198/3239	248283/29204
French	fra	Romance	32347/3232	780289/77416
Galician	glg	Romance	2472/1260	76208/36450
German	deu	Germanic	13814/799	229204/10727
Gothic	got	Germanic	3387/985	35024/10114
Greek	ell	Greek	1662/403	38139/9404
Hebrew	heb	Semitic	5241/484	122122/10050
Hindi	hin	Indic	13304/1659	262389/32850
Hungarian	hun	Ugric	910/441	17282/9974
Indonesian	ind	Malayo-Sumbawan	4477/559	82963/10676
Irish	gle	Celtic	121/445	2864/9554
Italian	ita	Romance	17427/1070	329477/18790
Japanese	jpn	Japanese	7164/511	145240/10404
Korean	kor	Korean	27410/3016	312830/32849
Latin	lat	Latin	30598/2568	387236/29858
Latvian	lav	Baltic	4124/989	51562/10773
Lithuanian	lit	Baltic	153/55	2536/883
Marathi	mar	Indic	373/46	2447/342
Norwegian	nob	Germanic	29870/4639	432741/62802
Old Church Slavonic	chu	Slavic	4123/1073	37432/10100
Persian	pes	Iranian	4798/599	110345/14474
Polish	pol	Slavic	6100/1027	52445/8613
Portuguese	por	Romance	17995/1770	401487/37388
Romanian	ron	Romance	8664/752	170551/14898
Russian	rus	Slavic	52664/7163	773678/105285
Serbian	srp	Slavic	2935/465	57581/8825
Slovak	slk	Slavic	8483/1060	65044/10648
Slovenian	slv	Slavic	7532/1817	106904/22083
Spanish	spa	Romance	28492/3054	731920/79171
Swedish	swe	Germanic	7041/1416	102400/23585
Tamil	tam	Sourthern Dravidian	400/80	5664/1118
Telugu	tel	South-Central Dravidian	1051/131	3926/519
Turkish	tur	Southwestern Turkic	3685/975	31271/8203
Ukrainian	ukr	Slavic	4506/577	61011/8384
Urdu	urd	Indic	4043/552	103152/13888
Vietnamese	vie	Viet-Muong	1400/800	17325/9873

Table S9: Languages with ISO codes, families (according to `https://universaldependencies.org/`), and the number of available sentence and words.