

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

Michael Hahn Daniel Jurafsky
Department of Linguistics Department of Linguistics
Stanford University Stanford University

Richard Futrell
Department of Language Science
University of California, Irvine

May 3, 2019

1 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 [1992].¹ Greenberg’s original study was based on 30 languages; more recently, Dryer [1992] documented the word order correlations based on typological data from 625 languages. Dryer [1992] 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

¹Regarding recent objections by Dunn et al. [2011], see follow-ups by Levy and Daumé [2011], and Croft et al. [2011].

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 to 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 *lifted_mark* relation. Third, we both the verb-PP relation and the relation between adjectives and their standard of comparison is captured by the *obl* relation.

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 Dryer [2013], 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 [1992] 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.

The resulting operationalization is shown in Table 1.

verb	Correlates with... object	Operationalization	Greenberg [1963]
adposition	NP	case	3, 4
copula verb	predicate	cop	–
‘want’	VP	xcomp	15
tense/aspect auxiliary	VP	aux	16, 13
negative auxiliary	VP		–
complementizer	S	mark	–
adverbial subordinator	S		–
noun	genitive	nmod	2, 23
noun	relative clause	acl	24
adjective	standard of comparison	obl	–
verb	PP		22

Significance levels: *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$

Table 1: Greenbergian Correlations according to Dryer Dryer [1992], with operationalizations with Universal Dependencies. We also provide the corresponding numbers in Greenberg’s original study.

2 Supporting Analyses for Study 1

Other tests, maybe som viz

Pareto Curve It has to be convex, so don’t need to do more.

3 Results on all UD Relations

Table 2 shows 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.

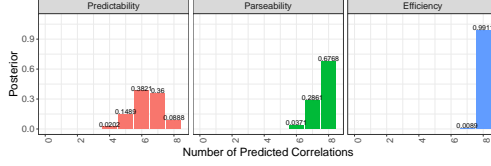


Figure 1: Posterior of the number of correlations correctly predicted by efficiency and its components, in a Bayesian multivariate mixed-effects logistic regression with random effects for languages and language families.

Further, we report results for grammars optimized for dependency length minimization (DLM). DLM can be seen as a heuristic approximation to efficiency. We also found that DLM is predicted by efficiency optimization. Indeed, we show that optimizing for DLM makes predictions similar to efficiency optimization.

4 Posterior

5 Relation to the Information Bottleneck

Our efficiency objective (Equation ??), to be maximized for word order grammars L , is restated below.

$$J_T(L) = I[L(T); T] - \lambda H[L(T)]. \quad (1)$$

This equation can be seen as a special case of the objective function proposed in Tishby et al. [1999] for general lossy compression, based on rate-distortion theory [Cover and Thomas, 2006, Harremoës and Tishby, 2007]. In that theory, the goal is to derive an encoding $\hat{X} = L(X)$ of some variable X which preserves the information in X relevant to some third variable Y , while minimizing irrelevant information. The three variables Y, X , and \hat{X} are assumed to form a Markov chain $Y \rightarrow X \xrightarrow{L} \hat{X}$. Given those variables, the information bottleneck objective to be maximized is given in Eq. 2.

$$J_{X,Y}(L) = I[\hat{X}; Y] - \lambda I[\hat{X}; X]. \quad (2)$$

In this equation, the joint distribution of X and Y is seen as fixed, and the conditional distribution $L = \hat{X}|X$ is optimized.

Relation	Real	DepL	Pred	Pars	Efficiency	Expected Prevalence
acl	80	85***	43	82***	71***	> 50% [Dryer, 1992]
aux	12	26***	19***	49	35**	< 50% [Dryer, 1992]
lifted_case	86	81***	40**	77***	67***	> 50% [Dryer, 1992]
lifted_cop	94	80***	59	72***	63**	> 50% [Dryer, 1992]
lifted_mark	76	85***	57*	76***	72***	> 50% [Dryer, 1992]
nmod	80	82***	49	71***	68***	> 50% [Dryer, 1992]
obl	88	78***	71***	46	68***	> 50% [Dryer, 1992]
xcomp	88	90***	76***	89***	85***	> 50% [Dryer, 1992]
nsubj	33	29***	53	7***	23***	> 50% [Dryer, 1992]
advcl	86	84***	51	67***	67*	> 50% [Greenberg, 1963, Diessel, 2001]
ccomp	86	88***	70**	77***	77**	> 50% (cf. Dryer [1980])
csubj	79	78***	60	66***	68**	> 50% (cf. Dryer [1980])
amod	51	84***	42	53*	51	≈ 50% [Dryer, 1992]
nummod	37	74**	49	52	52	≈ 50% [Dryer and Haspelmath, 2013, 89A, 83]
appos	77	77***	49	67***	62**	Unknown
lifted_cc	71	83***	51	81***	71***	Unknown
expl	12	36	62	40	53	Unknown
iobj	74	49	78***	38*	62	Unknown
vocative	46	42*	59	45	45*	Unknown
compound	66	61*	48	58*	52	Uninterpretable
det	31	71***	56*	48	54	Uninterpretable
dislocated	47	61	68**	55	58	Uninterpretable
dep	56	64*	52	58	58*	Uninterpretable
advmod	35	51	18***	47	39**	Uninterpretable
conj	72	82***	42**	62***	59*	UD artifact
discourse	24	36**	41*	49	46	UD artifact
fixed	78	32***	45	47	44	UD artifact
flat	73	66**	45	59**	54	UD artifact
goeswith	61	57	44	57	52	UD artifact
list	94	59	50	54	52	UD artifact
orphan	82	70**	50	55*	59	UD artifact
parataxis	66	56	43	57	59	UD artifact
reparandum	18	53	57	43	48	UD artifact

Significance levels: *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$

Table 2: Predictions on all UD relations occurring in more than one language. Numbers are the fraction of languages where the direction of the given relation agrees in direction with the *obj* relation. In the last column, we indicate what direction would be expected typologically. The first section correspond to universals from Dryer5 “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*).

	Correlates with...		Real	Efficiency	Predictability	Parseability
	verb <i>wrote</i>	object <i>letters</i>				
①	adposition <i>to</i>	NP <i>a friend</i>				
②	copula <i>is</i>	NP <i>a friend</i>				
③	auxiliary <i>has</i>	VP <i>written</i>				
④	noun <i>friend</i>	genitive <i>of John</i>				
⑤	noun <i>books</i>	relative clause <i>that you read</i>				
⑥	complementizer <i>that</i>	S <i>she has arrived</i>				
⑦	verb <i>went</i>	PP <i>to school</i>				
⑧	want <i>wants</i>	VP <i>to leave</i>				

Table 3: OPTION 3: Efficiency optimization accurately predicts the Greenbergian correlations. Each correlation is stated in terms of a pair of a ‘verb patterner’ and an ‘object patterner’, whose relative order correlate with that of verbs and objects. For each correlation, we provide an English example, and show the distribution among the actual grammars of the 51 languages (left), and among grammars optimized for efficiency (right). Blue bars count grammars conforming to the correlation, red bars count grammars violating it. Efficiency optimization predicts all eight correlations to hold in the majority of grammars.

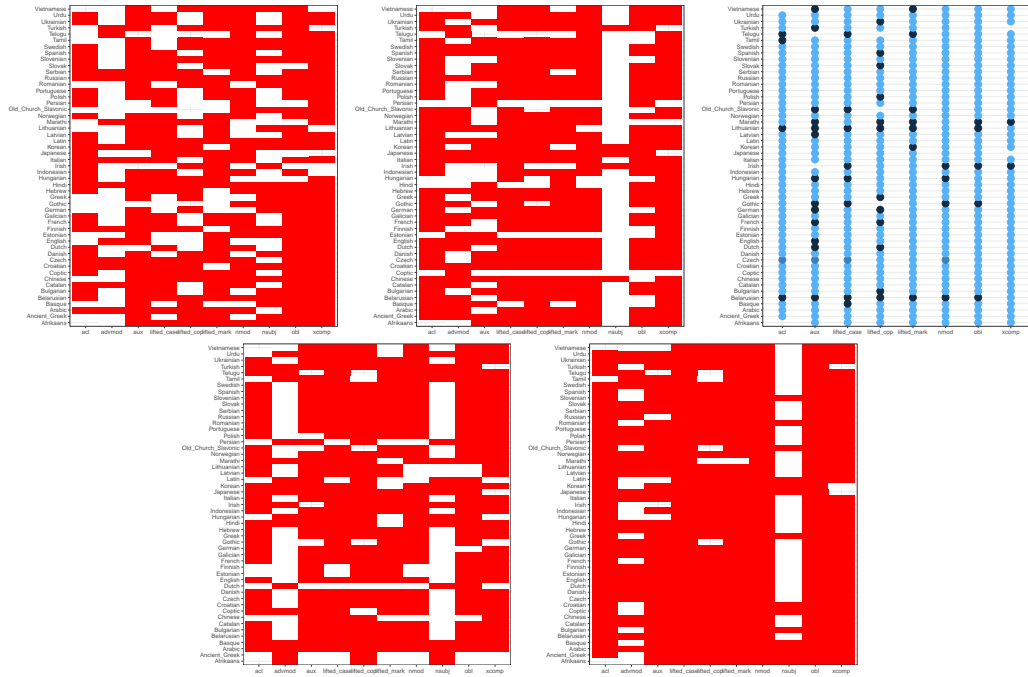


Figure 2: Correlations satisfied by the best word order grammars for predictability, parseability, efficiency, by the UD languages, and by the best grammars for dependency length.

In our setting we consider a language L expressing trees T as utterances U : $T \xrightarrow{L} U$. To make our formulation more parallel to the information bottleneck, we assume the utterances U are passed through a channel from producer to comprehender, and received as $\hat{U} = U$. We have the Markov chain $T \xrightarrow{L} U \rightarrow \hat{U}$, where we are assuming the conditional channel distribution $\hat{U}|U$ is fixed and the conditional distribution $L = U|T$ is to be optimized. Then our objective is parallel to Eq. 2:

$$J_T(L) = I[U; T] - \lambda I[U; \hat{U}]. \quad (3)$$

Assuming the channel $U \rightarrow \hat{U}$ is noiseless, we have $U = \hat{U}$, and thus $I[U; \hat{U}] = I[U; U] = H[U]$. Then Eq. 3 reduces to a form of Eq. 1:

$$J_T(L) = I[U; T] - \lambda H[U].$$

Our efficiency objective is thus closely related to the information bottleneck; the difference is that we are optimizing the mapping from underlying meanings to representations (strings) assuming that these strings will be transmitted in a fixed noiseless channel.

6 Possible values of λ

In the efficiency objective (1), the value of λ is constrained to be in $[0, 1]$. This means, surprisal must be weighted less strongly than parsability.

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

$$I[L(T); T] \leq H[L(T)]. \quad (4)$$

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

$$J_T(L) = I[L(T); T] - \lambda H[L(T)] \leq 0. \quad (5)$$

and it takes the maximally possible value zero if $L(T)$ maps all trees to a single utterance, in which case both $I[L(T); T]$ and $H[L(T)]$ 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 our experiments, we chose $\lambda = 0.9$ as a mathematically valid value that puts similar weight on both predictability and parsability. We previously carried out language optimization for $\lambda = 1$, and obtained essentially identical predictions for word order correlations and relative optimality of language.

The computational cost of grammar optimization precluded repeating the experiment for many values of λ . We note that each of the eight correlations is predicted by at least parseability or predictability, without any contradictory predictions (TODO check).

7 Joint optimization of grammar, parser, and language model parameters

7.1 Differentiable Ordering Grammars

In order to carry out optimization of grammars, we extend the space of grammars to 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 optimization with stochastic gradient descent, as we describe in Section 7.3.

The parameters of such a differentiable word order grammar are 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 $\{s_1, \dots, s_n\}$ placed on one side of a head, their order outward from the head is determined by iteratively sampling from the distribution $\text{softmax}(b_{\tau_1}, \dots, b_{\tau_n})$ (Goodfellow et al. [2016], 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 .

7.2 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 $\text{logit}(a_\tau) \sim \mathcal{N}(0, 1)$, $b_\tau \sim \mathcal{N}(0, 1)$. We then extract the posterior means for each parameter a_τ, b_τ , and convert the resulting differentiable grammar into an ordinary ordering grammar.

7.3 Optimizing Grammars for Efficiency

rewrite this

For each of our objective functions, we seek parameters θ of a word order grammar L_θ that maximize the average utility of sentences in the language obtained by linearizing unordered trees $t \sim T$ according to L_θ . The calculation of the predictability and parseability objectives also requires fitting language models parameterized by ϕ and parsers parameterized by ψ , respectively; and the calculation of the efficiency objective requires joint optimization of the word order grammar, language models, and parsers. The optimization problems come out to:

$$\max_{\theta} \mathbb{E}_{t \sim T} \mathbb{E}_{\mathbf{w} \sim L_\theta(t)} R_{DepL}(t; \mathbf{w}) \quad (6)$$

$$\max_{\theta, \phi} \mathbb{E}_{t \sim T} \mathbb{E}_{\mathbf{w} \sim L_\theta(t)} R_{Pred}^{\phi(\theta)}(t; \mathbf{w}) \quad (7)$$

$$\max_{\theta, \psi} \mathbb{E}_{t \sim T} \mathbb{E}_{\mathbf{w} \sim L_\theta(t)} R_{Pars}^{\psi(\theta)}(t; \mathbf{w}) \quad (8)$$

$$\max_{\theta, \phi, \psi} \mathbb{E}_{t \sim T} \mathbb{E}_{\mathbf{w} \sim L_\theta(t)} R_{Efficiency}^{\psi(\theta), \phi(\theta)}(t; \mathbf{w}). \quad (9)$$

Note that the original word orders from the treebank never enter this objective—in particular, they do not influence the language model and the parser, which are entirely determined by θ .

The case of dependency length in Eq. ?? is the simplest, requiring only coordinate descent on the parameters $a_\tau, b_\tau \in \theta$. A similar optimization problem was solved by hill-climbing in Gildea and Temperley [2007] and Gildea and Jaeger [2015]. We optimize this objective by stochastic gradient descent with a batch size of one unordered dependency tree.

The case of predictability, parseability, and efficiency are more challenging, as we need to simultaneously look for an ordering model optimized for

the loss of a language model and/or a parser, and for a language model and/or parser that are optimized for that given ordering model. We solve Eqs. 7–9 by performing stochastic gradient descent over all sets of parameters θ , ϕ , and/or ψ simultaneously. Below we discuss the process only for the predictability objective (Eq. 7); the other objectives are optimized analogously. In each 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 $\mathbf{w} \sim P_{L_\theta}(\cdot|t)$. Then we do a gradient descent step using the estimator

$$\begin{pmatrix} \partial_\theta (\log P_{L_\theta}(\mathbf{w}|t)) \cdot R_{Pred}^\phi(t; \mathbf{w}) \\ \partial_\phi R_{Pred}^\phi(t; \mathbf{w}) \end{pmatrix} \quad (10)$$

for the gradient

$$\begin{pmatrix} \partial_\theta \\ \partial_\phi \end{pmatrix} \mathbb{E}_{\mathbf{w}' \sim L_\theta(t)} R_{Pred}^\phi(t; \mathbf{w}').$$

This unbiased estimator is the ordinary gradient estimator for ϕ , and the REINFORCE estimator for θ [Williams, 1992].

In addition, we estimated maximum-likelihood ordering grammars on the original ordered dependency trees with SGD. For nonprojective trees, we ignore discontinuities.

A different commonly considered information-theoretic objective function for word order is Uniform Information Density [Jaeger, 2010], formalized as minimizing a superlinear function of per-word surprisal [Levy, 2018]. Note that optimizing grammars for this objective faces additional challenges, as the optimization problem cannot be rewritten as joint optimization in the form (7). We thus do not evaluate this objective function here.

8 Neural Network Architectures

In this section, we describe the details of the neural network architectures. All choices described in this section were fixed before evaluating word order properties and the efficiency of real language.

Estimating Predictability We choose a standard LSTM [Hochreiter and Schmidhuber, 1997] language model with vocabulary size restricted to the most frequent 50,000 words in the treebanks for a given language. Given the small 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. These decisions were made before evaluating word order properties.

Estimating Parseability We use a biaffine attention parser architecture [Kiperwasser and Goldberg, 2016, Zhang et al., 2017, Dozat et al., 2017]. 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 j th word is the head of the i th 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 [McDonald et al., 2005].

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 model had converged; parsing from POS tags prevents early overfitting. This decision was made before evaluating word order properties.

Hyperparameters **TODO study 1 only has 1 set of parameters**

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. 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 (predictability or parseability) on the language. We then used this setting for creating optimized word order grammars.

TODO random seeds for efficiency

All word and POS embeddings are randomly initialized with uniform values from $[-0.01, 0.01]$. We do not use pretrained embeddings [Peters et al., 2018]: 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.

Optimization details We employ two common variance reduction methods to improve the estimator (10), while keeping it unbiased. For predictability, note that the loss incurred for a specific word only depends on ordering decisions made up to that word (and its head, in the case of dependency length). 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. [2015] to obtain a version of (10) with lower variance. Second, we use a word-dependent moving average of recent per-word losses as control variate Williams [1992]. 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 [Xu et al., 2015] 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.

These techniques for improving (10) are well-known in the machine learning literature, and we fixed these before evaluating optimized grammars for word order properties.

We update word order grammar parameters θ using Stochastic Gradient Descent with momentum. For the language model parameters ϕ , we use plain SGD. For the parser parameters ψ , we use Adam [Kingma and Ba, 2014], following Dozat et al. [2017]. The learning rates and other optimization hyperparameters were determined together with the other hyperparameters.

For each language and objective function, we apply Early Stopping [Prechelt, 1998] as follows. We compute a Monte-Carlo estimate of the objective on the UD development set in intervals of 50,000 training steps, and stop optimization once the value of the objective deteriorates compared to the last previous estimate. This technique, well known in the machine learning literature, helps prevent overfitting to the specific corpora used.

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

Recall that, 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.

9 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 [Sirignano and Cont, 2018] and purely theoretical chaotic dynamical systems [Ogunmolu et al., 2016]; the neural graph-based parser is simply solving a minimal spanning tree problem [McDonald et al., 2005]. 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 prediction and parseability 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 a particular prediction and parsing algorithms.

9.1 CKY Parsers

We constructed simple Probabilistic Context-Free Grammars (PCFGs) from corpora and word order grammars, using a simplified version of the models of Collins 1997 (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:

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

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.

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 parsing loss $H[T|L_\theta(T)]$ on the validation partition of the English UD corpora, for random and optimized ordering grammars. Results (Figure 3) show that optimized grammars are more parseable for exact parsing of a simple PCFG.

9.2 Distorted graph-based parsers

Many of the word order biases explained by parseability in Table ?? are also explained by dependency length minimization. Therefore, we address here the idea that the graph-based parser might have a built-in bias toward parses involving short dependencies, which we call a **locality bias**.

A simple way to eliminate or alter a locality bias for any parser would be to change the order in which the parser sees words. Suppose that a parser P

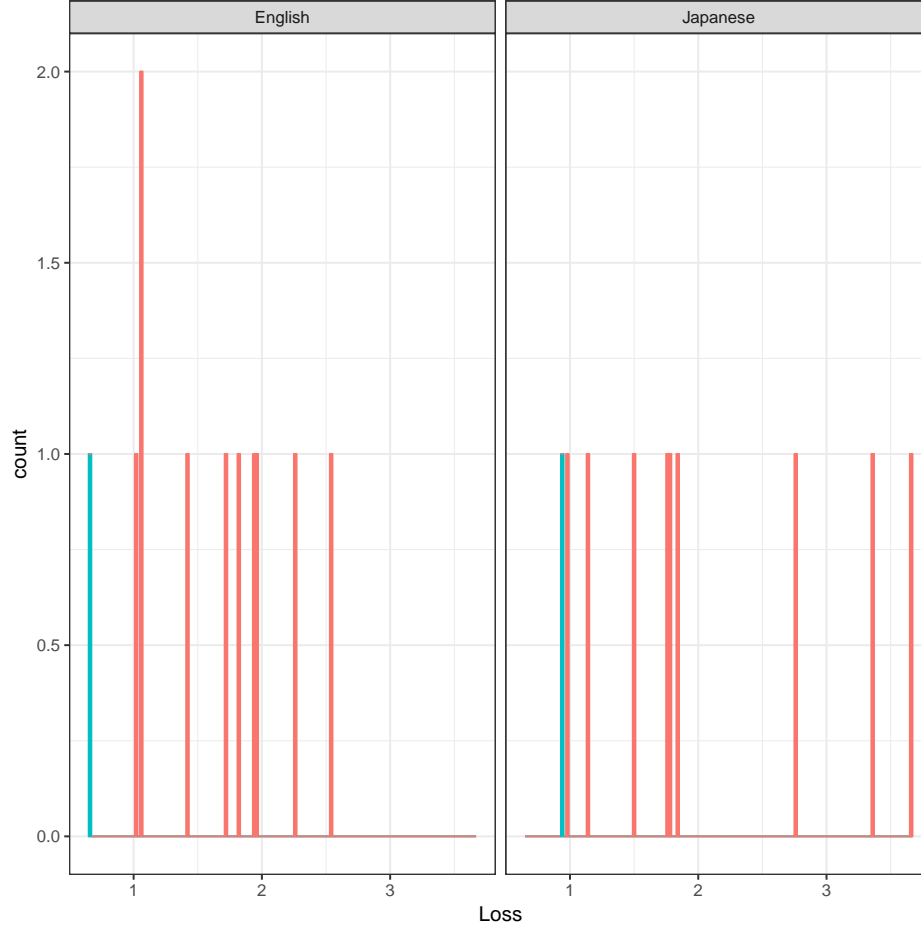


Figure 3: Parsability loss $H[T|L_\theta(T)]$ (lower is better) computed by a simple CKY parser, for random word order grammars (red) and word order grammars optimized for efficiency (blue).

has a bias toward positing a dependency between a word and the immediate previous word.

Even-odd order. A sequence of n words originally ordered as $w_1w_2w_3w_4 \cdots w_n$ is reordered by separating the even and odd indices: $w_2w_4w_6 \cdots w_{n-1}w_1w_3w_5 \cdots w_n$ (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 $w_1w_2w_3 \cdots w_n$ is split in half at the middle (index $m = \lceil n/2 \rceil$), and the two resulting sequences are interleaved, yielding $w_1w_mw_2w_{m+1}w_3w_{m+3} \cdots w_n$. 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 $w_1w_2w_3 \cdots w_{n-1}w_n$ is ordered from the edges of the string inwards, as $w_1w_nw_2w_{n-1} \cdots w_{\lceil n/2 \rceil}$. 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.

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 4. 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.

9.3 n-gram language models

We constructed a Bigram model with Kneser-Ney smoothing. 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 5) show that languages optimized for efficiency are optimal for a bigram language model.

9.4 Discussion

It is always a logical possibility in studies such as this one that the results are dependent on the particular probabilistic models used.

10 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

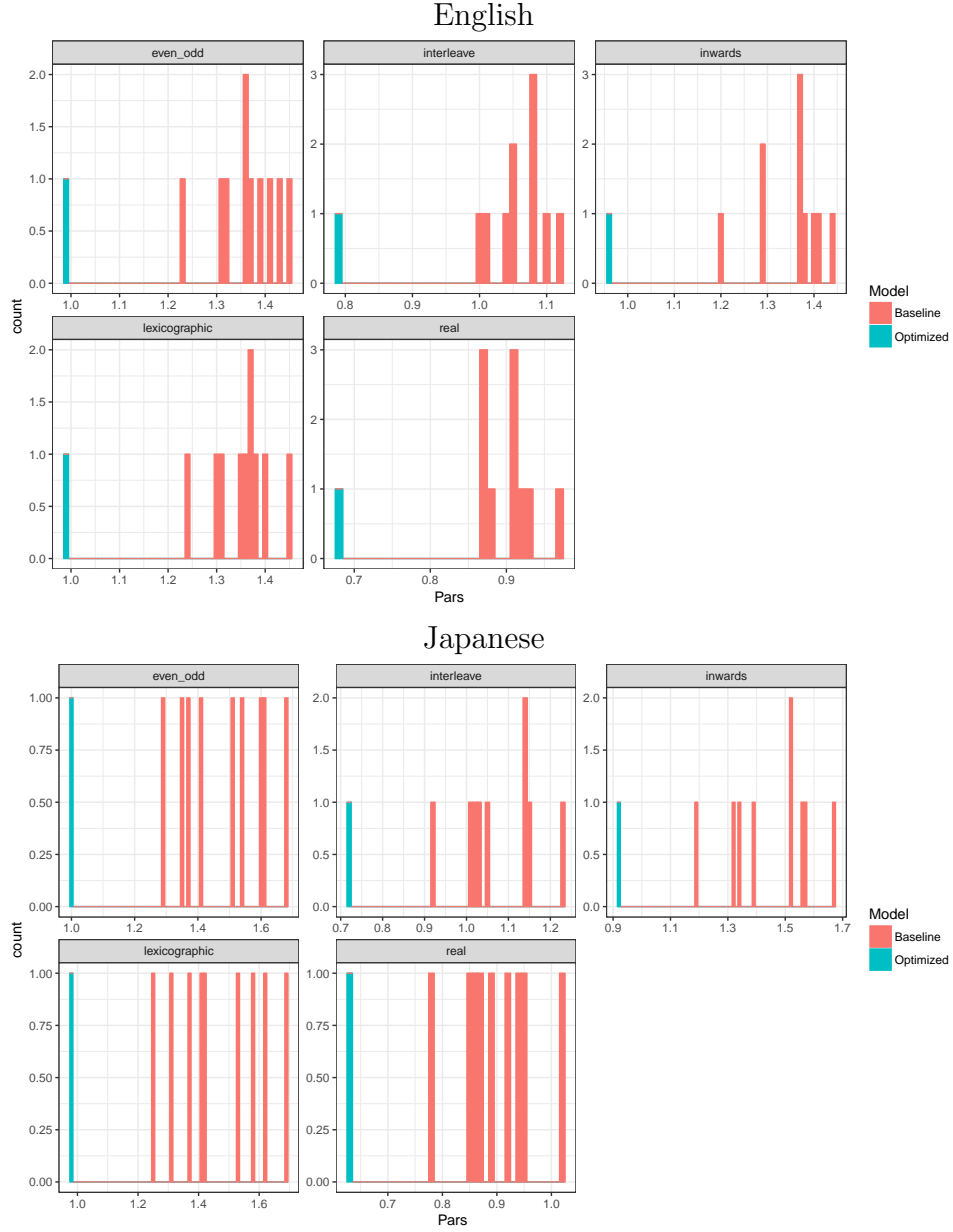


Figure 4: Parseability of baseline grammars and grammars optimized for efficiency, in English (top) and Japanese (bottom), measured by parsing loss $H[T|L_\theta(T)]$ (lower is better), for the four distorted orderings, and the actual orderings ('real').

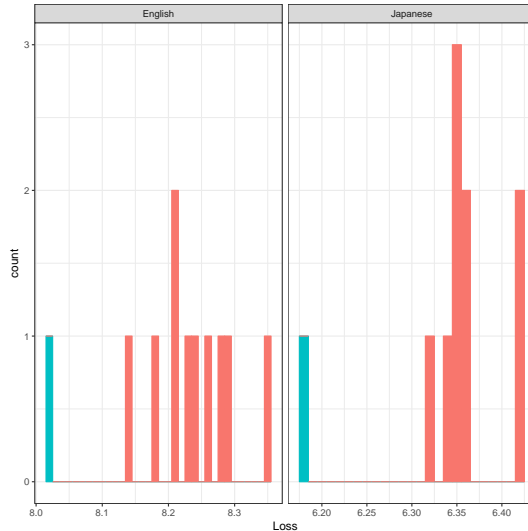


Figure 5: Surprisal (lower is better) computed from Bigram model, on English and Japanese data ordered according to random ordering grammars (red) and ordering grammars optimized for efficiency (blue).

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 ≈ 300 times more data than is available for some other UD languages. We considered both a random ordering grammar, and the best ordering grammar optimized for parseability. 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.

do same for language model, and for the other random grammars

The resulting curves are shown in Figure 6. A gap in parsing accuracy of about 0.07-0.1 appears already at 0.01 % of the training data (2000 words), and persists for larger amounts of training data.

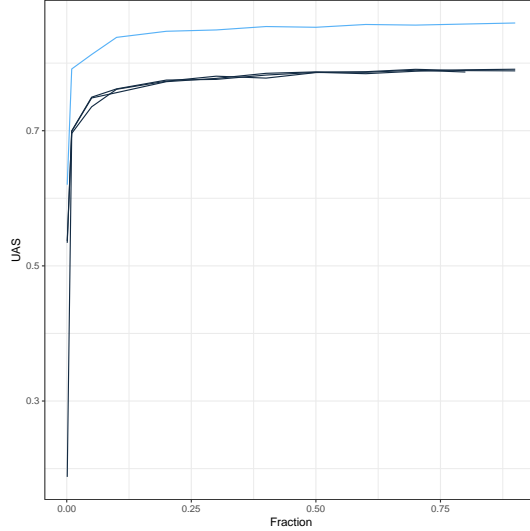


Figure 6: Parsing accuracy (measured in UAS) for optimized (light blue) and random (black) ordering grammar on Czech data, as a function of the fraction of total training data provided.

11 Study 2, Correlation between Universals and Efficiency

12 Comparing Extracted Grammars to Linguistic Judgments

Here, we validate our grammars by comparing the dominant orders of six syntactic relations that are also annotated in WALS. Among the eight relations that we were able to test, five are annotated in WALS. Here, we compare our grammars with WALS on these five relations, and the verb-object relation.

WALS has some data for all but seven languages.² WALS has data for 74% of the entries, and lists a dominant order for 91% of these. The grammars we extracted from the corpora agree with WALS in 96 % out of those cases where WALS records a dominant word order.

²Note that 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 agrees with the Serbian grammar.

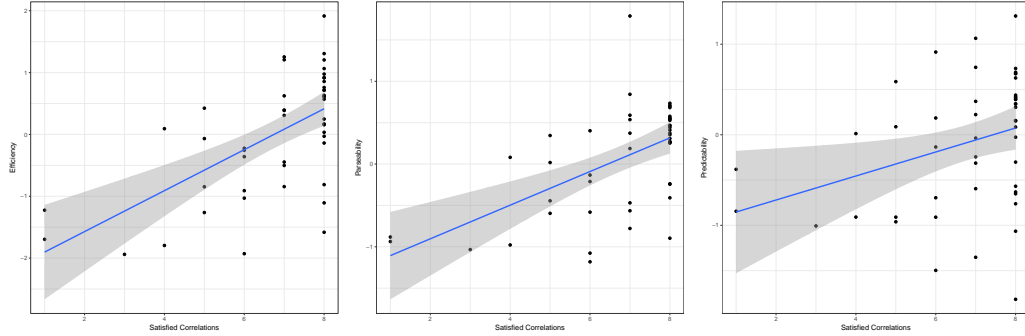


Figure 7: Correlation between the number of satisfied correlations (x -axis) and efficiency, parseability, and predictability (y -axis), for the 51 real languages.

For each of the six syntactic relation, the first column provides the order 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 in the language, or that insufficient data was available when compiling WALS [Dryer, 2011]. Finally, ‘–’ indicates that the relation does not occur in the corpus.

Language	Objects		Adpositions		Compl.		Rel.Cl.		Genitive		PP	
Afrikaans	DH	?	HD	?	HD	?	–	?	HD	?	HD	?
Anc.Gr.k.	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
Galician	HD	?	HD	?	HD	?	HD	?	HD	?	?
German	HD	*	HD	HD	HD	HD	HD	HD	HD	DH	*
Gothic	HD	?	HD	?	HD	?	HD	?	HD	?	?
Greek	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	?
Hebrew	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	?
Hindi	DH	DH	DH	DH	HD	HD	DH	*	DH	DH	DH
Hungarian	<i>DH</i>	HD	DH	DH	HD	HD	HD	*	DH	DH	DH
Indonesian	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD
Irish	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD
Italian	HD	HD	HD	HD	HD	HD	HD	HD	HD	HD	?
Japanese	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH
Korean	DH	DH	DH	DH	<i>HD</i>	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
Swedish	HD	HD	HD	HD	HD	HD	HD	HD	<i>HD</i>	DH	HD
Tamil	DH	DH	DH	DH	DH	*	<i>HD</i>	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
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	<i>DH</i>	HD	—	HD	HD	HD	HD

References

- Matthew S Dryer. The Greenbergian word order correlations. *Language*, 68(1): 81–138, 1992.
- M. Dunn, S.J. Greenhill, S.C. Levinson, and R.D. Gray. Evolved structure of language shows lineage-specific trends in word-order universals. *Nature*, 473 (7345):79–82, 2011.
- Roger Levy and Hal Daumé. Computational methods are invaluable for typology, but the models must match the questions. *Linguistic Typology*, 15(2):393–399, 2011.
- William Croft, Tanmoy Bhattacharya, Dave Kleinschmidt, D Eric Smith, and T Florian Jaeger. Greenbergian universals, diachrony, and statistical analyses. *Linguistic Typology*, 15(2):433–453, 2011.
- 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>.
- 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.
- Holger Diessel. The ordering distribution of main and adverbial clauses: A typological study. *Language*, pages 433–455, 2001.
- 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.
- Matthew S. Dryer and Martin Haspelmath. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig, 2013. URL <https://wals.info/>.
- Naftali Tishby, Fernando Pereira, and William Bialek. The information bottleneck method. In *Invited paper to The 37th annual Allerton Conference on Communication, Control, and Computing*, 1999.
- Thomas M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Hoboken, NJ, 2006.

- Peter Harremoës and Naftali Tishby. The information bottleneck revisited or how to choose a good distortion measure. In *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pages 566–570. IEEE, 2007.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- Daniel Gildea and David Temperley. Optimizing grammars for minimum dependency length. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 184–191, Prague, Czech Republic, June 2007. URL <http://www.aclweb.org/anthology/P07-1024>.
- Daniel Gildea and T. Florian Jaeger. Human languages order information efficiently. *arXiv*, 1510.02823, 2015. URL <http://arxiv.org/abs/1510.02823>.
- Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.
- T. Florian Jaeger. Redundancy and reduction: Speakers manage syntactic information density. *Cognitive Psychology*, 61(1):23–62, 2010.
- Roger P. Levy. Communicative efficiency, uniform information density, and the rational speech act theory. In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society*, page 684–689, 2018.
- Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- 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.
- 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.
- 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.

- 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.
- 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.
- 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.
- 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.
- Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference Learning Representations*, 2014.
- Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.
- Justin Sirignano and Rama Cont. Universal features of price formation in financial markets: perspectives from deep learning. *arXiv preprint arXiv:1803.06917*, 2018.
- Olalekan Ogunmolu, Xuejun Gu, Steve Jiang, and Nicholas Gans. Nonlinear systems identification using deep dynamic neural networks. *arXiv preprint arXiv:1610.01439*, 2016.
- Matthew S Dryer. The evidence for word order correlations. *Linguistic Typology*, 15(2):335–380, 2011.