Proceedings of SALT 35: 286–309, 2025

Quantifiers that are more monotone are easier to learn

Christopher Haberland
University of Washington

Shane Steinert-Threlkeld
University of Washington

Abstract Linguistic universals have been hypothesized to bound the observed properties of natural languages at all levels of linguistic analysis (Greenberg 1966; Croft 2002; van der Hulst 2008). In the domain of semantics, such universals are often explained by appeal to general information-theoretic and cognitive facts, from efficient communication to ease of learning. This paper expands the range of the ease of learning explanation through a case study of monotone quantifiers. While most existing studies compare expressions which either do or do not satisfy a given universal property, we here define a metric for measuring monotonicity in *degrees*. A computational experiment shows that this degree correlates well with ease of learning, providing a finer-grained explanation in the ease of learning tradition.

Keywords: semantic universals, generalized quantifiers, monotonicity, neural networks, learnability

1 Introduction

Linguistic *universals* are properties that are hypothesized to limit the form of all natural languages (Greenberg 1966; Comrie 1989; Croft 2002). A subset of universals include *semantic universals*, which are those universals that are thought to define the bounds of semantic systems across all (or nearly all) natural human languages. Since von Fintel & Matthewson's (2008) survey on the topic that noted a dearth of work on understanding the natural causes of semantic universals, a line of investigation has opened to uncover evidence for proposed explanations of semantic universals. A key question beyond merely uncovering new universals is understanding the pressures that lead to their emergence. Among these explanations, *ease of learning* has been proposed as an influence on the distribution of semantic universals observed in natural languages (Chemla et al. 2019b; Steinert-Threlkeld & Szymanik 2020, 2019; Maldonado et al. 2022).

The present paper expands and enriches an ease of learning explanation for a particular universal, by generalizing the universal to one that can be measured in degrees. In particular, we focus on the universal that all simple determiners in natural languages are monotonic (Barwise & Cooper 1981; Keenan & Stavi 1986). We design a computational experiment to uncover evidence for ease of learning

as an explanation of this monotonicity universal. Our experiment expands beyond past work (Steinert-Threlkeld & Szymanik 2019) that finds preliminary evidence for monotonicity affecting ease of quantifier learning among neural agents—that monotone quantifiers are easier to learn than minimally different non-monotone ones—by substantially broadening the number of quantifiers tested and measuring their monotonicity in *degrees*, generalizing the method described in the work of Carcassi et al. (2021) and Steinert-Threlkeld (2021). Our experiments discover a general correlation between ease of learning in neural network training experiments and this degree. This correlation holds despite the fact that the logical complexity (in a sense made precise below) of the tested quantifiers is mostly constant.

In Sections 2–4, we summarize foundational work on semantic universals and models of learning that motivate our investigation. In Section 5, we describe how we employ these concepts to study the relationship of monotonicity and learning of quantifier expressions and present and discuss our experimental results in Sections 6–7.

2 Explaining Semantic Universals

Initial approaches to discovering semantic universals have historically focused on typological comparison of language to gather initial evidence for the existence of semantic universals. More recently, typological and quantitative approaches have consolidated support for the existence of semantic universals. Examples providing empirical evidence of semantic universals are numerous, covering many facets of natural language, including lexical classification systems (Youn et al. 2016), spatial demonstratives (Diessel 2014), color (Berlin & Kay 1991), properties of indefinite pronouns (Haspelmath 2001), and properties of quantifiers (Bach et al. 1995; Keenan & Paperno 2012; Paperno & Keenan 2017).

As a next step, computational approaches have sought to shed light on exactly why semantic universals arise. One type of explanation argues that universals maximize the efficiency of communication by optimizing an inherent trade-off between *simplicity* and *informativeness*. Simplicity measures ease of mental representation, while informativeness describes the amount of successfully communicated information (Kemp et al. 2018; Gibson et al. 2019). The efficiency argument has been supported by experimental evidence of a host of proposed universals, including in the domains of kinship (Kemp & Regier 2012), color categories (Regier et al. 2007; Zaslavsky et al. 2018), indefinite pronouns (Denić et al. 2021), modals (Imel & Steinert-Threlkeld 2022; Steinert-Threlkeld et al. 2023), logical connectives (Uegaki 2023; Enguehard & Spector 2021; Bar-Lev & Katzir 2025), and quantifiers (Steinert-Threlkeld 2021).

While these studies may offer domain-specific measures of simplicity in an at-

tempt to measure it, a generalized metric is elusive. The study of *ease of learning* as an explanation for universals evades this problem by taking the perspective that simplicity may be revealed simply through the course of learning a concept; a simple concept will be learned more quickly than a complex concept. Steinert-Threlkeld & Szymanik (2019) synthesize two different proposals put forth to explain why universals arise in relation to their ease of learning. The first argument made by Barwise & Cooper (1981) and Keenan & Stavi (1986) argue that universals take hold because they necessarily decrease the hypothesis space for a learner. Studies show, however, that this alone does not necessarily make the learning problem significantly easier (Piantadosi et al. 2012). If this were the only reason universals arose, it would entail that synchronic entrenchment of random linguistic choices arising from repeated communication would be the primary cause of universals, a process emphasized in usage-based theory (summarized in Christiansen et al. (2009)).

The second argument has it that certain properties are inherently easier to learn and are therefore more likely to be lexicalized. The focus of this hypothesis descends from Chomsky (1965) and the stance that biological constraints determine a Universal Grammar, but unlike Chomsky's assertions, does not emphasize innate faculties as a source for the emergence of universals. This perspective is epitomized by Feldman (2000) in early study that measured human concept learning as tracking with logical complexity, concluding that "human conceptual difficulty reflects intrinsic mathematical complexity after all, rather than some idiosyncratic and uniquely human bias." This view is echoed by Chemla et al. (2019a), observing that learnability of universals may depend on factors that are common in any learning process, whether enacted by humans or machine architectures, regardless of particular biases embedded in the learning agent.

Steinert-Threlkeld & Szymanik (2019) provide supportive evidence in this vein by measuring the time to convergence for neural networks trained on quantifiers that adhere to various universal properties that are lexicalized in all simple determiners, including monotonicity, quantity, and conservativity. In paired trials of verifying quantifier expressions adhering to each universal or not adhering to each universal, the neural model was quicker to converge for the quantifiers satisfying universal properties. Their empirical results are the first to support the idea that quantifier expressions satisfying universals in and of themselves are easier to learn. This style of explanation has been extended to color terms (Steinert-Threlkeld & Szymanik 2020; Douven forthcoming), responsive predicates (Steinert-Threlkeld 2020; Maldonado et al. 2022), and contrafactives (Strohmaier & Wimmer 2022, 2025). Rule learning experiments in humans (Chemla et al. 2019a) and baboons (Chemla et al. 2019b) have also uncovered a preference for monotone concepts.

3 (Degrees of) Monotonicity

To set the stage for introducing a *degree* of monotonicity, we first recall the traditional definition of a monotone quantifier. Informally, a quantifier is considered *monotone* when its truth value is preserved under set inclusion in one (or both) of its arguments. Formally, following the tradition in generalized quantifier theory (Barwise & Cooper 1981; Peters & Westerståhl 2008; Szymanik 2016), let $\langle M, A, B\rangle$ be a model where $A, B \subseteq M$. Write $Q(A,B)$ for the truth value returned by quantifier Q on that model, and let the primed variables A', B' denote *alternative* sets obtained by enlarging or shrinking the original ones. Then, we can define upward and downward monotonicity in each of the left and right arguments as in Table 1.

Flavor	**Monotonicity Condition**	**Example**
Up-Right	$Q(A,B) \wedge B \subseteq B' \Rightarrow Q(A,B')$	*Most Seattleites like to hike.* $\Longrightarrow$ *Most Seattleites like to go outdoors.*
Up-Left	$Q(A,B) \wedge A \subseteq A' \Rightarrow Q(A',B)$	*Some domestic cats meow.* $\Longrightarrow$ *Some felines meow.*
Down-Right	$Q(A,B) \wedge B' \subseteq B \Rightarrow Q(A,B')$	*Not all planets are bigger than Earth.* $\Rightarrow$ *Not all planets are bigger than Saturn.*
Down-Left	$Q(A,B) \wedge A' \subseteq A \Rightarrow Q(A',B)$	*All stars shine.* $\Longrightarrow$ *All dwarf stars shine.*

Table 1 Four flavors of monotonicity with their formal definitions and illustrative examples.

According to the above definitions, being monotone is a binary property: a quantifier either is or is not monotone. Intuitively, however, quantifiers can be *more* and *less* monotone. For example, "between m and n" and "an even number of" are both non-monotone. The former, however, is in some sense 'closer' to being monotone: there is a continuous set of numerical values for which "between 3 and 8 people ate a plantain yesterday" is true. By contrast, whether or not "an even number of people ate a plantain yesterday" has no such pattern: its truth-value fluctuates constantly across this numerical spectrum.[1]

To capture this intuition, we generalize the *degree of monotonicity* of Steinert-Threlkeld (2021) and model the property of monotonicity as a continuous feature. This allows us to capture greater variation in the manifestation of monotonicity that

1 We note that "between m and n" is also a conjunction of monotone quantifiers, whereas the latter is not, and so is "connected" in the sense of Chemla et al. (2019a). Our measure below does not directly correspond to this logical sense of complexity, a point to which we return in the discussion.

can aid in uncovering the effect of this property on other factors, such as ease of learning, with higher resolution.

With $f : \langle A, \leq_A \rangle \to \langle B, \leq_B \rangle$ an arbitrary function between partially ordered sets, we define $f^{\uparrow}$, the minimal monotone extension of f, by $f^{\uparrow}(x) = \sup\{f(x') : x' \leq_A x\}$. (Note: this definition requires that suprema exist in B.) This definition has a few nice properties: (i) $f^{\uparrow}$ is monotone; (ii) f is monotone iff $f = f^{\uparrow}$; (iii) $f^{\uparrow}$ is the 'least' monotone extension of f: if $f \leq g$ (where this means that $f(x) \leq g(x) \forall x \in A$) and g is monotone, then $f^{\uparrow} \leq g$. For the present application of this definition, f can be taken to be the characteristic function of a generalized quantifier with respect to the B argument. As an example, if f is the characteristic function of "between 3 and 5", the resulting minimal monotone extension would be the characteristic function ($f^{\uparrow}$) of "at least 3".

We then define: $\mathsf{mon}(f) := I(f; f^{\uparrow})/H(f)$, where I is the mutual information and H is the entropy. This measures how much uncertainty about the value of f is resolved by knowing the value of its minimal monotone extension $f^{\uparrow}$. From property (ii), it follows that $\mathsf{mon}(f) = 1$ *iff* f is monotone. For quantifiers: we apply this definition to the quantifier's indicator function on the right argument, in addition to the left argument, and then take the maximum value of those and a symmetric definition for downward monotonicity. Carcassi et al. (2021) report that 'some" incurred a degree of monotonicity value of 1.0, "between 3 and 5" a value of 0.7517, and an even number of registered a degree 0.001 in their implementation of this metric. These values capture the intuition about quantifiers being more and less monotone with which we began. More examples of calculated degrees of monotonicity for logical expressions used in our learning experiments are provided in Table 4 in Section 5.

4 Models of learning

To understand the relationship between a semantic property and ease of learning, it is necessary to define what learning is and how it may be measured. We provide background on two distinct research traditions that bear on our experimental design, including our choice for an implementation of a learning model, the creation and representation of the concepts on which we test that model, and our measure for ease of learning itself.

4.1 Connectionism

The *connectionist* literature over the past several decades reflects on the core elements that define a learning system, principally drawing on the human brain for inspiration (Medler 1998). Connectionist models of learning describe a class of

models that are principally 1. composed of layers of 2. homogeneous, primitive computational units that are 3. recursively updated during a "learning" process (Hanson & Burr 1990).

Artificial neural network (ANN) models are the most ubiquitous and widely implemented models of learning that are used today and were inspired by neuronal activation in the brain (Gurney 1997). In ANNs, a series of computational units, or neurons, are joined together into a larger computational layer, and updated to approximate a function $f(x)$ that maps input data x to output data observations y. Each neuron consists of a weight w and bias term b that linearly map an input value to an output value. Stacking additional neurons together in parallel or in succession can lead to "deep" networks of greater power to model an observed output (LeCun et al. 2015). The backpropagation algorithm (Rumelhart et al. 1986) is the foundational mechanism by which the network propagates a signal proportional to a loss function and change the weights of its component neurons.

Although deep neural networks have been posited to loosely model learning in the brain, the degree to which they do so remains an open question. The field may attempt answer this question by comparing the representations of information stored by backpropagation and resulting distribution of responses from neural models, though current methods for analyzing and measuring the brain and how it propagates information are still developing (Saxe et al. 2021). Exactly how backpropagation mirrors the brain's assignment of credit to neuronal layers to assist in forming distributed representations remains unclear, and indeed, other models of credit assignment that update the representations have been proposed to be more brain-like. Song et al. (2024) have proposed an alternative learning mechanism to evade backpropagation's problem of 'catastrophic forgetting' problem and avoid its tendency to require orders of magnitude more data than the human brain to create useful distributed representations of a concept.

Regardless of the exact learning algorithm used, recent work has argued that the foundation of backpropagation in accumulating synaptic change through modulating feedback connections reflect the core mechanism by which the human cortex learns (Lillicrap et al. 2020), supporting the use of an implementation of learning distinct from the human brain for our experiments. That neural networks 1. automatically learn directly from "sensory" input observations to conduct "behavioral" tasks (Doerig et al. 2023), 2. internalize abstract representations by learning higher-order functions over inputs (Templeton et al. 2024), and 3. have been shown to have the capacity to match humans in learning efficiently over a data distribution justifies them as suitable models of learning in the abstract: With respect to this later point, Hosseini et al. (2024) have shown that neural networks are able to at least predict human brain responses to language on a similar scale of training data as would be experienced by a human during development of linguistic abilities, suggesting that

the lower bound of competence achievable by neural networks for similar scales of learned inputs may be comparable for defined domains.

Portelance & Jasbi (2024) provide a framework for conceptualizing the neural network as a model for modeling language acquisition, and cognitive processing more broadly. This work identifies a particular class of studies that utilize neural networks as models with limited cognitive assumptions, which do not commit to casting neural networks as models of cognition. In our work, neural networks can be thought to model at least one aspect of learning systems that could in principle also apply to human learning, even if the process of training a neural network does not exactly model the learning process conducted by the human brain.

4.1.1 Neural Architectures

This work compares two different architectures of neural models: the LSTM and the Transformer. The LSTM (Long Short-Term Memory) (Hochreiter & Schmidhuber 1997) network is a type of RNN (Recurrent Neural Network), a class of state-sequential neural models. A given state in a classic RNN network (Elman 1990) depends upon the state calculated at a previous timestep, which makes the model sensitive to order in sequences. RNNs are dependent upon prior states' cells:

$$\mathbf{s}_t = g(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{sh}\mathbf{s}_{t-1} + \mathbf{b}_h)$$

where $\mathbf{s}_t$ is an output state at a particular timestep in the model, $\mathbf{W}_{xh}$ is the current state weight matrix, and $\mathbf{W}_{sh}$ is the hidden matrix that depends on the state of the last cell. g is a nonlinear activation function and $\mathbf{b}_h$) is a bias term (Goldberg 2017). Most of the practical advances in deep learning utilizing RNN architectures have been attained by the LSTM (Yu et al. 2019), which introduces memory gates to dividing each sequence state in half, allowing separate channels for "long-term memory" and working memory in the network. Long-term memory cells are dependent upon gate designated for selectively reducing or amplifying signals from past states. The gating mechanism in the LSTM allows current states to be sensitive to information across longer sequential horizons than in the tradition RNN architecture.

The Transformer (Vaswani et al. 2017) is a more recently developed sequence-processing neural architecture that is based on the attention mechanism originally described in Bahdanau et al. (2015). With the attention mechanism, learned representations adapt by learning which parts of sequences are most useful (and should be *attended* to) to produce an output. Self-attention is used in the Transformer, in which attention layers are added to an encoder to allow the model to better learn associations between different components of the input sequence. This is achieved

by calculating key, query, and value vectors that create separate representations of combinations of particular inputs in the sequence.

4.2 Language of Thought

Our work implements a "Language of Thought" (LoT) (Fodor 1975, 2008; Schneider 2011) grammar to generate a wide range of quantifiers. The Language of Thought Hypothesis posits that thinking involves manipulation of distinct, structured mental representations called symbols. An important feature of an LoT is that combining component symbols can lead to *productivity*, or an infinite number of outputs by conjoining symbols according to the LoT grammar. Connecting the outputs of licit combinations of mental symbols together forms higher level concepts or functions. The Language of Thought Hypothesis is philosophically aligned with compositional, symbolic computational approaches to learning in its assumption that structured representations of primitive concepts compose to form higher cognitive functions. Quilty-Dunn et al. (2022) argue that the Language of Thought program remains the strongest model for explaining cognitive processes, by summarizing a LoT's key components and providing supporting evidence from linguistics, perception, system-1 reasoning, and animal cognition. Yang & Piantadosi (2022) assert that the LoT provides strong parallels to cognition across various domains and may demonstrate rich potential for testing the theoretical limits of learning systems.

A large body of work has been devoted to bridging symbolic and connectionist approaches to learning (Besold et al. 2021) on the basis that elements of both theories likely make up the structures that power human intelligence (Piantadosi 2021; Smolensky et al. 2022). Both approaches have inspired working implementations of learning reliant on the use of statistical updating to change the model in response to a target distribution of data (Piantadosi 2021; Yang & Piantadosi 2022).

We use the Python package ULTK (Imel et al. 2025) to define a grammar composed of primitive functions that may be composed to build larger quantifier expressions. The primitive functions are shown in Table 2. While we adopt set and boolean logical operators utilized by van de Pol et al. (2023) to constitute the LoT grammar used in our experiments, we advise the reader to refer to Piantadosi et al. (2016) for a fuller discussion of the affect of the choice of primitives on learning within different domains.

5 Experiments

We model our experiment after Steinert-Threlkeld & Szymanik (2019) and frame the learning task as *quantifier sentence verification*. In this setup, a neural network

operator	type	gloss
$\cup$	$\text{SET} \times \text{SET} \to \text{SET}$	Union
$\cap$	$\text{SET} \times \text{SET} \to \text{SET}$	Intersection
$\setminus$	$\text{SET} \times \text{SET} \to \text{SET}$	Difference
$i(\cdot,\cdot)$	$\text{INT} \times \text{SET} \to \text{SET}$	"Referent at index"
$\lvert\cdot\rvert$	$\text{SET} \to \text{INT}$	Cardinality
$\subseteq$	$\text{SET} \times \text{SET} \to \text{BOOL}$	Subset equal
$=$	$\text{INT} \times \text{INT} \to \text{BOOL}$	Integer equality
$>$	$\text{INT} \times \text{INT} \to \text{BOOL}$	Integer greater than
$\neg$	$\text{BOOL} \to \text{BOOL}$	Negation
$\wedge$	$\text{BOOL} \times \text{BOOL} \to \text{BOOL}$	And
$\vee$	$\text{BOOL} \times \text{BOOL} \to \text{BOOL}$	Or

Table 2 The grammar of operators used to generate quantifiers.

undergoes supervised learning, or is shown a series of examples, to internalize a decision function that distinguishes inputs as either verified by a given quantifier or not. The inputs we provide to the neural network are series of ordered indices representing a set of referents in a discourse, where each referent can belong to ordered sets M, A, or B, as framed in generalized quantifier theory.

We conduct two sets of learning experiments that are distinguished by the chosen neural learning architecture. For experiments (a), an LSTM architecture is utilized. For experiment (b), an autoregressive Transformer architecture is employed. Hyperparameter decisions were made to ensure that both the LSTM and Transformer learners had a roughly equivalent number of total parameters (9k and 12.7k, respectively). For each model architecture, we conduct two trials to better understand the natural variance between trials of different initializations. Therefore, trials (a) and (b) for each experiment are simply repeated trials of the same experiment with distinct seeds for sampling the training data and training the neural weights. In each trial, we randomly select 2000 expressions generated from the `ultk` grammar at a depth of 5, and a maximum length of 4 for the number of elements in an input model. These generated expressions are licit combinations composed of functions in a defined grammar of set, numeric, and boolean operations, up to a certain operation depth ($d = 5$). We choose the depth and length parameters to balance competing desires: we aimed to test a large variety of possible quantifiers that are

guaranteed to be minimal in logical complexity and non-duplicative in meaning, but needed a feasible generation procedure against a large combinatorial space of possible quantifiers.

For a given training run, we draw a quantifier from a random sample of expressions generated in the aforementioned procedure. For each quantifier, we generate training data by randomly instantiating model triples constrained by two conditions: 1) the maximum limit to the number of indices pertaining to the set M, where $A \subseteq M, B \subseteq M$, is fixed at m, and 2) the cardinality of the model input x. We generate 10000 models total for each quantifier, split evenly by whether the quantifier model is verified by the given expression. We generate up to 1000 batches of 1000 training examples in each batch, and record the entropy of the class distribution of each batch. If the entropy of a generated batch is less than .02, or if 5000 examples for either class (verified or not verified by the quantifier) are not found after 1000 iterations, the expression is not tested in subsequent experiments. This mitigates the effect of class imbalance on our learning experiments.

We use a binary cross entropy loss function for all experiments which combines the sigmoid and binary cross entropy functions in a single layer. The Adam optimizer with a fixed learning rate of 0.001 was used for all experiments. We tested using variable learning rates and the AdamW (Loshchilov & Hutter 2017) optimizer that is popular for training transformers, but did not find a significant benefit for improving the model's learnability of expressions. For this reason, we ran all experiments with the same learning rate and optimizer, despite common discrepancies in the practice of training of LSTMs and Transformers.

Training a neural network requires transforming the problem space into a representative encoding. Each input is encoded as a vector that is the concatenation of a binary encoding for each referent in the model triple. The encoding of each model referent, or index, denotes its inclusion in A, B, and M. For example, a referent that pertains to A is encoded as $[1,0,1]$, and a referent included only in B is encoded as $[0,1,1]$. The '1' in the third position of each of these encodings reflects that sets A and B necessarily intersect M. As an example, let Ref_1 belong to A and B, Ref_2 to M, and Ref_3 to no set. Then, the input encoding would be:

$$encode(Ref_{1;\in AB}, Ref_{2;\in M}, Ref_{3;\notin M}) = [1,1,1,0,0,1,0,0,0]$$

The supervised learning paradigm requires us to also encode whether the model verifies the quantifier or not. Therefore, each triple is paired with a value Y that records either 1 (*True*) or 0 (*False*) according to whether $Q(A,B) = 1$ for the target quantifier Q. The neural model outputs a probability $\hat{Y}$ that $Q(A,B) = 1$ and the learning algorithm adjusts the model's weights so that $\hat{Y}$ approximates Y over training time. We include referents that do not pertain to any of the sets to encourage the neural network to further generalize the quantifier function to novel orientations and

Algorithm 1 Sample_Expressions(Q, x, B, L)

function SAMPLE_EXPRESSIONS(Q, x, B, L)
  $\mathscr{S}_{\text{true}} \leftarrow \emptyset, \quad \mathscr{S}_{\text{false}} \leftarrow \emptyset$
  iter $\leftarrow 0$
  while true **do**
    batch $\leftarrow$ GENERATEBATCH(Q, x, B) ▷ draw B random triples
    for all $(M, A, B) \in$ batch **do**
      if $Q(A, B) = 1$ **then**
        $\mathscr{S}_{\text{true}} \leftarrow \mathscr{S}_{\text{true}} \cup \{(M, A, B)\}$
      else
        $\mathscr{S}_{\text{false}} \leftarrow \mathscr{S}_{\text{false}} \cup \{(M, A, B)\}$
      end if
    end for
    iter $\leftarrow$ iter $+1$
    $H \leftarrow \text{Entropy}\big(|\mathscr{S}_{\text{true}}|, |\mathscr{S}_{\text{false}}|\big)$
    if $|\mathscr{S}_{\text{true}}| \geq L \wedge |\mathscr{S}_{\text{false}}| \geq L \wedge H \geq 0.02$ **then**
      break
    else if $H < 0.02$ **or** iter ≥ 1000 **then**
      error("Insufficient balance")
    end if
  end while
  return Shuffle$\big(\text{MergeSelections}(\mathscr{S}_{\text{true}}, \mathscr{S}_{\text{false}}, L)\big)$
end function

to discount absolute positions of indices in the ordered input data while learning.

Our procedure yielded training runs for roughly 90% of the 2000 samples expressions for each of our trials. The majority of expressions were of a depth of 4: see Table 3 for the average counts of training runs by expression depth in each trial. The quantifiers that were not initialized for training were discarded for having training inputs of insufficient entropy of truth values ($H < .02$) or not attaining a sufficiently balanced training set after 1000 random batches. Such quantifiers are almost uniformly true or false across generated models, and so do not provide a strong learning signal for the minority class. For each quantifier, we also calculated the degree of monotonicity measure. See Table 4 for examples of this measure.

5.1 Ease of Learning Metric

Our chosen measurement of ease of learning is the area-under-the-curve of the validation loss of the neural model. This is the sum of the validation loss recorded

Expression Depth	Count
1	2
2	7
3	101
4	1683

Table 3 Average Number of Training Runs by Expression Depth across Trials 1a - 2b

Form (English)	Expression	$mon(Q)$ Interpretations			
		R-U	**L-U**	**R-D**	**L-D**
"all"	$A \subseteq B$	**1.0**	0.0	0.0	**1.0**
"not all"	$\neg\big(A \subseteq B\big)$	.059	**1.0**	**1.0**	.059
"most"	$\lvert A \cap B\rvert > \lvert A \setminus B\rvert$	**1.0**	.148	.012	.287
"some"	$\neg\big(A \cap B = B \setminus B\big)$	**1.0**	**1.0**	.059	.059
"some but not all"	$\neg\big(A \cap B = B \setminus B\big) \wedge \neg\big(A \subseteq B\big)$	.491	**1.0**	.491	.054
N/A	$\neg\big(A \subseteq B\big) \wedge \big(\big\lvert A \cap B\big\rvert > \big\lvert A \setminus B\big\rvert\big)$	**.734**	.571	.256	.194
N/A	$\neg\big(\lvert B\rvert = \lvert A \setminus B\rvert\big)$	.001	.001	0.0	**.2**
N/A	$\big(A \subseteq B\big) \lvert \big(B \subseteq A\big)$	**0.0**	**0.0**	**0.0**	**0.0**

Table 4 Quantifier expressions with their corresponding degree of monotonicity values ($M = 6,\ X = 6$). R-U: right-upward; L-U: left-upward; R-D: right-downward; L-D: left-downward

during each batch during training. Let $L(t)$ denote the validation loss at training step (or time) t. Then, the area under the curve (AUC) over the interval $[T_0, T_1]$ is defined as

$$\mathrm{AUC}_{[T_0,T_1]} = \int_{T_0}^{T_1} L(t)\,dt.$$

If the validation loss is recorded at discrete time steps $t_1, t_2, \ldots, t_N$ with a constant time interval Δt, the AUC can be approximated by

$$\mathrm{AUC}_{[T_0,T_1]} \approx \Delta t \sum_{i=1}^{N} L(t_i).$$

This metric does not need to be normalized, as all model runs are over the same time period.

The validation loss AUC allows us to consider a greater number of records for analysis as all training runs that were not excluded from training due to the sampling conditions have values for the AUC metric. That is, for any panel of completed runs (all runs for a particular expression), the AUC metric is calculable. This is in contrast to a metric recording the earliest step for which a training run converged according to a defined loss threshold, as some runs may not converge within the allotted training budget of 50 epochs. As such, the AUC may provide a more precise proxy for learning in our experimental setup. The aggregate validation loss metric also offers another conceptual improvement over a time-to-convergence metric: two expression training runs could have equal convergence times, but the run of one run might accrue sufficiently higher loss during the training period. Intuitively, this expression is more difficult to learn, yet a time-to-convergence metric is not able to capture such variation.

We find our measurement of aggregate validation loss AUC to be highly correlated with a time-to-convergence metric[2] (see Figure 1). Fewer Transformer model runs converged within the training budget compared to the basic LSTM model. Table 5 shows that we are able to obtain a metric for nearly equal numbers of runs across training experiments by measuring ease of learning as aggregate validation loss, as opposed to a step of convergence.

Experiment	Model	Training started	Training converged
1a	LSTM	1790	1263
1b	LSTM	1792	1262
2a	Transformer	1794	775
2b	Transformer	1791	763

Table 5 The number of runs for which training started was variable due to the conditions that precluded training defined in Algorithm 1. We define convergence as obtaining an average validation loss below .05 over the previous 50 batches.

Code and data for reproducing these experiments, including full hyper-parameter details, may be found within the official GitHub repo[3] for the Python package `ultk`, described in detail in Imel et al. (2025).

2 This metric is the first training step after which a training run as having converged when the validation loss of a training run minimizes sufficiently to stay below .05 for 20 successive batches of training examples.

3 https://github.com/CLMBRs/ultk/tree/main/src/examples/learn_quant

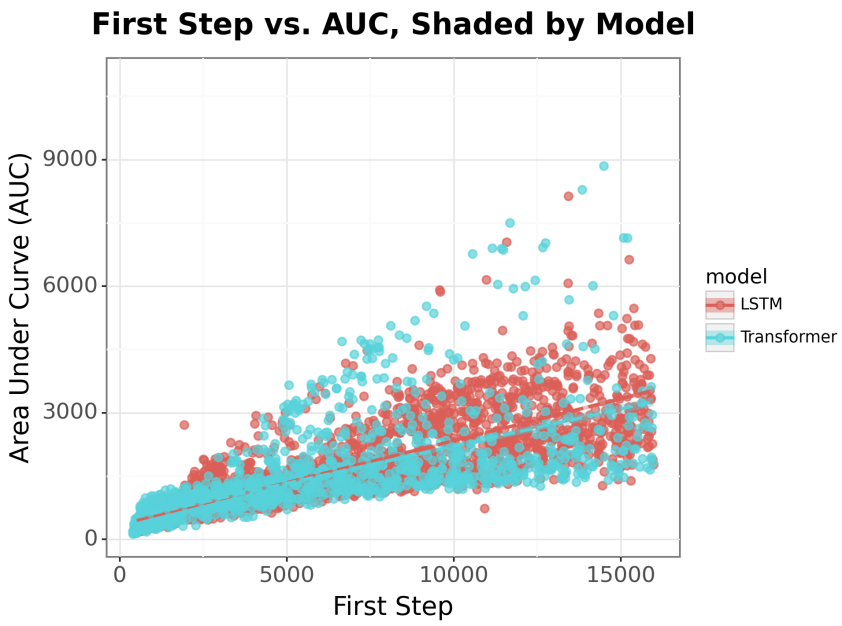

Figure 1 Partial Spearman correlation, controlled for *model*: $(r = 0.892)$.

6 Results

Figure 2 shows the degree of monotonicity calculated for each quantifier expression plotted against each run's summed validation loss for all steps during training. Visual inspection reveals a wide disparity in the model's ability to learn different quantifier expressions. A Pearson coefficient (a measure of linear dependence) reveals an inverse correlation (-0.3453; $p <= 8.22 \times 10^{-200}$) between degree of monotonicity and validation loss AUC, supporting the hypothesis that more monotone quantifiers are easier to learn.

We fit a linear mixed effects model that attempts to account for model choice (whether a Transformer or LSTM) while measuring the effect of monotonicity on aggregate validation loss. These choices reflect our desire to know whether the neural model architecture helped determine the rate of learning. A separate question is whether, while accounting for differences observed between the models, there was an additional interaction effect between the monotonicity of the learned expression and the model choice in predicting the rate at which the expression would be learned by the model. Expressions were modeled with random effects, where each individual expression is allowed a random intercept.

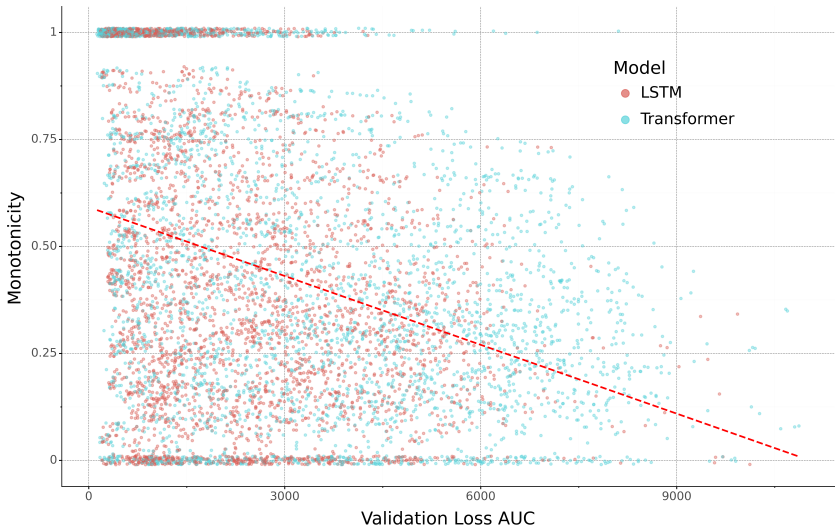

Figure 2 Monotonicity vs summed AUC values for the validation loss for all steps during training for all runs. LSTM runs (1a, 1b) and Transformer runs (2a, 2b) are shown in red and blue, respectively.

$$\begin{aligned}\text{ValLossAUC}_{ij} &= \beta_0 + \beta_1\left(\textit{Transformer}_{ij}\right) + \beta_2\left(\textit{Monotonicity}_{ij}\right) \\ &\quad + \beta_3\left(\textit{Transformer}_{ij} \times \textit{Monotonicity}_{ij}\right) + u_{0i} + e_{ij}, \\ u_{0i} &\sim N(0, \sigma_u^2), \quad e_{ij} \sim N(0, \sigma^2).\end{aligned}$$

In this model, i indexes the *expression* (each expression is a random-effects group), and j indexes each *observation* for expression i (e.g., each trial or run). The dummy variable $\textit{Transformer}_{ij}$ equals 1 if the run employed the Transformer model and 0 for the LSTM model. $\textit{Monotonicity}_{ij}$ is a continuous measure of monotonicity for the given expression and run. The fixed-effects coefficients $\beta_0, \beta_1, \beta_2, \beta_3$ represent, respectively: (a) the baseline intercept (LSTM at Monotonicity $= 0$), (b) the effect of using Transformer versus LSTM at Monotonicity $= 0$, (c) the slope of monotonicity for the LSTM model, and (d) the additional interaction effect of monotonicity for the Transformer model. The random intercept u_{0i} allows each expression to shift the intercept up or down, with $u_{0i} \sim \mathcal{N}(0, \sigma_u^2)$. The residual error ε_{ij} for each observation (i, j) is assumed $\varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$.

The regression model uncovers a significant negative coefficient for the *Monotonicity* parameter. That is, a higher degree of monotonicity is inversely correlated with the aggregate validation loss and consequently there is a positive correla-

Parameter	Coefficient	Std. Error	p-value
Intercept	3403.436	95.966	<0.001
C(model)[T. Transformer]	1583.381	32.826	<0.001
Monotonicity	-1789.666	146.214	<0.001
C(model)[T. Transformer]: Monotonicity	-1058.114	50.009	<0.001

Table 6 Fixed Effects Estimates for the AUC of Validation Loss Steps

tion with ease of learning. The coefficient of the $Transformer$ factor is positive, and outweighs the magnitude of the interaction variable $Transformer : Monotonicity$, suggesting that the Transformer-based model was less effective overall in learning the task.

7 Discussion

Due to the exponential combinatorics of the LoT grammar, over 93% of the trials in our experimentally generated expressions were of the same complexity (i.e. had the same number of primitives). This suggests that despite a nearly constant measure of the complexity of the learned expressions, there is a positive relationship between monotonicity and ease of learning. We can conclude then that the effect of monotonicity that we discover cannot be reduced to an effect of logical complexity. Future work can explore in more detail the interaction effects of both complexity and monotonicity.

The interaction term between model choice and monotonicity suggests that the differential effect of monotonicity on ease of learning between the LSTM and Transformer models is attenuated at higher values of monotonicity. In other words: the LSTM's advantage in learning arises in the lower monotonicity regime. These findings do not necessarily uncover a comparative advantage between the Transformer and LSTM architectures when learning functions of varying monotonicity; it could be that with additional optimization of the model's parameters that model choice effects may be nullified. Future work could address potential biases of architectural choices by experimenting across a greater variety of architectures and refining the parameterization according to more exhaustive hyperparameter tuning regimes.

We note that the results of our experiments revealed special cases in which our chosen degree of monotonicity taken from Steinert-Threlkeld (2021) departed from intuition. Models that have a verified submodel assigned to no set, or whose supermodels belong to A and B for all referent indices may sometimes "saturate" this de-

gree of monotonicity metric, leading to a greater portion of expressions categorized as having degree 0 of monotonicity. (An example: $A \subseteq B|B \subseteq A$ receives degree 0 for the reason that its minimal monotone extension is the trivial (always true) quantifier.) Because many of these 'defective' quantifiers were learned relatively quickly by the neural learners, the effect is that the relationship between monotonicity and validation loss AUC is measured to be less than would be expected. That we still uncovered a significant inverse relationship between monotonicity and the validation loss AUC testifies to its pervasiveness across a diverse range of quantifiers. Future work may consider the comparative strengths of using different measures of monotonicity, such as the measure utilized by Zhu (2019), which is directly proportional to the number of "switches" or discontinuities between true and false values in the ordered set of referents, but is not as general as the current definition. It may also be possible to reduce our measure of degree to an LoT-based one with special primitives; we leave these connections for future work.

While we utilize LSTMs and Transformers under a supervised learning paradigm as our learning models, future work could consider other architectures and training paradigms. For example, reinforcement learning techniques have been argued to have significant parallels to the type of learning involved in human cognition (Collins 2019), and Bayesian inference methods have been argued to exhibit the necessary abstractions for general concept learning (Piantadosi et al. 2016).

8 Conclusion

Our study has advanced prior work demonstrating that monotone quantifiers are generally easier to learn than non-monotone ones by introducing a degree of monotonicity and showing that monotonicity is positively correlated with ease of learning. We note that most of our quantifiers have the same length of shortest formula, which shows that degree of monotonicity is correlated with ease of learning over and above what can be explained by minimum description length. This bolsters the case that ease of learning explains monotonicity and potentially other semantic universals. Future work will generalize further by using this monotonic degree metric to explain other empirical facts and by considering an even wider range of quantifiers.

References

Bach, Elke, Eloise Jelinek, Angelika Kratzer & Barbara H Partee (eds.). 1995. *Quantification in Natural Languages*, vol. 54. Springer. doi:10.1007/978-94-017-2817-1.

Bahdanau, Dzmitry, Kyunghyun Cho & Yoshua Bengio. 2015. Neural machine

translation by jointly learning to align and translate. In Yoshua Bengio & Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, doi:10.48550/arXiv.1409.0473. http://arxiv.org/abs/1409.0473.

Bar-Lev, Moshe E. & Roni Katzir. 2025. Communicative stability and the typology of logical operators. *Linguistic Inquiry* 56(3). 415–437. doi:10.1162/ling_a_00497.

Barwise, Jon & Robin Cooper. 1981. Generalized quantifiers and natural language. In *Philosophy, language, and artificial intelligence: Resources for processing natural language*, 241–301. Springer. doi:10.1007/BF00350139.

Berlin, Brent & Paul Kay. 1991. *Basic color terms: their universality and evolution*. University of California Press.

Besold, Tarek R, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luis C Lamb, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas et al. 2021. Neural-symbolic learning and reasoning: a survey and interpretation. In *Neuro-symbolic artificial intelligence: The state of the art*, 1–51. IOS press. doi:10.3233/faia210348. http://dx.doi.org/10.3233/faia210348.

Carcassi, Fausto, Shane Steinert-Threlkeld & Jakub Szymanik. 2021. Monotone quantifiers emerge via iterated learning. *Cognitive Science* 45(8). doi:10.1111/cogs.13027. http://dx.doi.org/10.1111/cogs.13027.

Chemla, Emmanuel, Brian Buccola & Isabelle Dautriche. 2019a. Connecting content and logical words. *Journal of Semantics* 36(3). 531547. doi:10.1093/jos/ffz001. http://dx.doi.org/10.1093/jos/ffz001.

Chemla, Emmanuel, Isabelle Dautriche, Brian Buccola & Joël Fagot. 2019b. Constraints on the lexicons of human languages have cognitive roots present in baboons (papio papio). *National Academy of Sciences* 116(30). 1492614930. doi:10.1073/pnas.1907023116. http://dx.doi.org/10.1073/pnas.1907023116.

Chomsky, Noam. 1965. *Aspects of the Theory of Syntax*. The MIT Press 50th edn. doi:10.21236/ad0616323. https://www.jstor.org/stable/j.ctt17kk81z.

Christiansen, Morten H., Chris Collins & Shimon Edelman. 2009. Language universals: a collaborative project for the language sciences. In Morten H. Christiansen, Chris Collins & Shimon Edelman (eds.), *Language Universals*, 0. Oxford University Press. doi:10.1093/acprof:oso/9780195305432.003.0001. https://doi.org/10.1093/acprof:oso/9780195305432.003.0001.

Collins, Anne Gabrielle Eva. 2019. Reinforcement learning: bringing together computation and cognition. *Current Opinion in Behavioral Sciences* 29. 6368. doi:10.1016/j.cobeha.2019.04.011. http://dx.doi.org/10.1016/j.cobeha.2019.04.011.

Comrie, Bernard. 1989. *Language Universals and Linguistic Typology: Syntax*

and Morphology. Chicago, IL: University of Chicago Press 2nd edn. https://press.uchicago.edu/ucp/books/book/chicago/L/bo24426144.html.

Croft, William. 2002. *Typology and Universals*. Cambridge University Press. doi:10.1017/cbo9780511840579. http://dx.doi.org/10.1017/cbo9780511840579.

Denić, Milica, Shane Steinert-Threlkeld & Jakub Szymanik. 2021. Complexity/informativeness trade-off in the domain of indefinite pronouns. *Semantics and Linguistic Theory* 30. 166. doi:10.3765/salt.v30i0.4811. http://dx.doi.org/10.3765/salt.v30i0.4811.

Diessel, Holger. 2014. Demonstratives, frames of reference, and semantic universals of space. *Language and Linguistics Compass* 8(3). 116–132. doi:10.1111/lnc3.12066. http://dx.doi.org/10.1111/lnc3.12066.

Doerig, Adrien, Rowan P. Sommers, Katja Seeliger, Blake Richards, Jenann Ismael, Grace W. Lindsay, Konrad P. Kording, Talia Konkle, Marcel A. J. van Gerven, Nikolaus Kriegeskorte & Tim C. Kietzmann. 2023. The neuroconnectionist research programme. *Nature Reviews Neuroscience* 24(7). 431–450. doi:10.1038/s41583-023-00705-w. https://www.nature.com/articles/s41583-023-00705-w.

Douven, Igor. forthcoming. The learnability of natural concepts. *Mind & Language* doi:10.1111/mila.12523. https://onlinelibrary.wiley.com/doi/abs/10.1111/mila.12523.

Elman, Jeffrey L. 1990. Finding structure in time. *Cognitive Science* 14(2). 179211. doi:10.1016/0364-0213(90)90002-e. http://dx.doi.org/10.1016/0364-0213(90)90002-e.

Enguehard, Émile & Benjamin Spector. 2021. Explaining gaps in the logical lexicon of natural languages: A decision-theoretic perspective on the square of Aristotle. *Semantics and Pragmatics* 14(5). doi:10.3765/sp.14.5. http://semprag.org/article/view/sp.14.5.

Feldman, Jacob. 2000. Minimization of Boolean complexity in human concept learning. *Nature* 407(6804). 630–633. doi:10.1038/35036586. https://www.nature.com/articles/35036586.

von Fintel, Kai & Lisa Matthewson. 2008. Universals in semantics. *The Linguistic Review* 25(1-2). doi:10.1515/TLIR.2008.004. https://www.degruyter.com/document/doi/10.1515/TLIR.2008.004/html.

Fodor, Jerry A. 1975. *The Language of Thought*, vol. 5. Harvard University Press. https://philarchive.org/archive/PORTLO-9.

Fodor, Jerry A. 2008. *LOT 2: The Language of Thought Revisited*. Oxford: Oxford University Press 1st edn. doi:10.1093/acprof:oso/9780199548774.001.0001. http://dx.doi.org/10.1093/acprof:oso/9780199548774.001.0001.

Gibson, Edward, Richard Futrell, Steven T. Piantadosi, Isabelle Dautriche, Kyle

Mahowald, Leon Bergen & Roger Philip Levy. 2019. How efficiency shapes human language doi:10.31234/osf.io/w5m38. http://dx.doi.org/10.31234/osf.io/w5m38.

Goldberg, Yoav. 2017. *Neural Network Methods for Natural Language Processing*. Springer International Publishing. doi:10.1007/978-3-031-02165-7. http://dx.doi.org/10.1007/978-3-031-02165-7.

Greenberg, Joseph H. 1966. *Language universals*, vol. 8. Mouton.

Gurney, Kevin. 1997. *An introduction to neural networks*. Taylor I& Francis. doi:10.4324/9780203451519. http://dx.doi.org/10.4324/9780203451519.

Hanson, Stephen José & David J. Burr. 1990. What connectionist models learn: Learning and representation in connectionist networks. *Behavioral and Brain Sciences* 13(3). 471–489. doi:10.1017/S0140525X00079760. https://www.cambridge.org/core/product/identifier/S0140525X00079760/type/journal_article.

Haspelmath, Martin. 2001. *Indefinite Pronouns*. Oxford University Press. doi:10.1093/oso/9780198235606.001.0001. http://dx.doi.org/10.1093/oso/9780198235606.001.0001.

Hochreiter, Sepp & Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9(8). 17351780. doi:10.1162/neco.1997.9.8.1735. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Hosseini, Eghbal A., Martin Schrimpf, Yian Zhang, Samuel Bowman, Noga Zaslavsky & Evelina Fedorenko. 2024. Artificial neural network language models predict human brain responses to language even after a developmentally realistic amount of training. *Neurobiology of Language* 5(1). 4363. doi:10.1162/nol_a_00137. http://dx.doi.org/10.1162/nol_a_00137.

van der Hulst, Harry. 2008. On the question of linguistic universals. *The Linguistic Review* 25(1-2). 1–34. doi:10.1515/TLIR.2008.001. https://www.degruyter.com/document/doi/10.1515/TLIR.2008.001/html.

Imel, Nathaniel, Christopher Haberland & Shane Steinert-Threlkeld. 2025. The Unnatural Language ToolKit (ULTK). *Society for Computation in Linguistics (SCiL)* 8(1). doi:10.7275/scil.3144. https://openpublishing.library.umass.edu/scil/article/id/3144/.

Imel, Nathaniel & Shane Steinert-Threlkeld. 2022. Modal semantic universals optimize the simplicity/informativeness trade-off. *Semantics and Linguistic Theory* 1. 227. doi:10.3765/salt.v1i0.5346. http://dx.doi.org/10.3765/salt.v1i0.5346.

Keenan, Edward L & Denis Paperno (eds.). 2012. *Handbook of Quantifiers in Natural Language*, vol. 90. Springer. doi:10.1007/978-94-007-2681-9.

Keenan, Edward L. & Jonathan Stavi. 1986. A semantic characterization of natural language determiners. *Linguistics and Philosophy* 9(3). 253326. doi:10.1007/bf00630273. http://dx.doi.org/10.1007/bf00630273.

Kemp, Charles & Terry Regier. 2012. Kinship categories across languages reflect general communicative principles. *Science* 336(6084). 1049–1054. doi:10.1126/science.1218811. https://www.science.org/doi/10.1126/science.1218811.

Kemp, Charles, Yang Xu & Terry Regier. 2018. Semantic typology and efficient communication. *Annual Review of Linguistics* 4(1). 109128. doi:10.1146/annurev-linguistics-011817-045406. http://dx.doi.org/10.1146/annurev-linguistics-011817-045406.

LeCun, Yann, Yoshua Bengio & Geoffrey Hinton. 2015. Deep learning. *Nature* 521(7553). 436444. doi:10.1038/nature14539. http://dx.doi.org/10.1038/nature14539.

Lillicrap, Timothy P., Adam Santoro, Luke Marris, Colin J. Akerman & Geoffrey Hinton. 2020. Backpropagation and the brain. *Nature Reviews Neuroscience* 21(6). 335346. doi:10.1038/s41583-020-0277-3. http://dx.doi.org/10.1038/s41583-020-0277-3.

Loshchilov, Ilya & Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* .

Maldonado, Mora, Jennifer Culbertson & Wataru Uegaki. 2022. Learnability and constraints on the semantics of clause-embedding predicates. *Annual Meeting of the Cognitive Science Society* 44(44). https://escholarship.org/uc/item/9h13v9db.

Medler, David A. 1998. A brief history of connectionism. *Neural computing surveys* 1. 18–72.

Paperno, Denis & Edward L Keenan (eds.). 2017. *Handbook of Quantifiers in Natural Language: Volume II*. Springer International Publishing. doi:10.1007/978-3-319-44330-0. http://dx.doi.org/10.1007/978-3-319-44330-0.

Peters, Stanley & Dag Westerståhl. 2008. *Quantifiers in language and logic*. Oxford University Press. doi:10.1093/acprof:oso/9780199291267.001.0001. https://doi.org/10.1093/acprof:oso/9780199291267.001.0001.

Piantadosi, Steven, Joshua Tenenbaum & Noah Goodman. 2016. The logical primitives of thought: empirical foundations for compositional cognitive models. *Psychological Review* 123(4). 392424. doi:10.1037/a0039980. http://dx.doi.org/10.1037/a0039980.

Piantadosi, Steven T. 2021. The computational origin of representation. *Minds Mach.* 31(1). 1–58. doi:10.1007/s11023-020-09540-9. https://doi.org/10.1007/s11023-020-09540-9.

Piantadosi, Steven T., Joshua B. Tenenbaum & Noah D. Goodman. 2012. Modeling the acquisition of quantifier semantics: a case study in function word learnability https://api.semanticscholar.org/CorpusID:29918589.

van de Pol, Iris, Paul Lodder, Leendert van Maanen, Shane Steinert-

Threlkeld & Jakub Szymanik. 2023. Quantifiers satisfying semantic universals have shorter minimal description length. *Cognition* 232. 105150. doi:10.1016/j.cognition.2022.105150. https://www.sciencedirect.com/science/article/pii/S001002772200138X.

Portelance, Eva & Masoud Jasbi. 2024. The roles of neural networks in language acquisition. *Language and Linguistics Compass* 18(6). doi:10.1111/lnc3.70001. http://dx.doi.org/10.1111/lnc3.70001.

Quilty-Dunn, Jake, Nicolas Porot & Eric Mandelbaum. 2022. The best game in town: the re-emergence of the Language of Thought Hypothesis across the cognitive sciences. *Behavioral and Brain Sciences* 1–55. doi:10.1017/S0140525X22002849.

Regier, Terry, Paul Kay & Naveen Khetarpal. 2007. Color naming reflects optimal partitions of color space. *National Academy of Sciences* 104(4). 14361441. doi:10.1073/pnas.0610341104. http://dx.doi.org/10.1073/pnas.0610341104.

Rumelhart, David E., Geoffrey E. Hinton & Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323(6088). 533–536. doi:10.1038/323533a0. https://www.nature.com/articles/323533a0.

Saxe, Andrew, Stephanie Nelli & Christopher Summerfield. 2021. If deep learning is the answer, what is the question? *Nature Reviews Neuroscience* 22(1). 55–67. doi:10.1038/s41583-020-00395-8. https://www.nature.com/articles/s41583-020-00395-8.

Schneider, Susan. 2011. *The Language of Thought: A New Philosophical Direction*. The MIT Press. doi:10.7551/mitpress/9780262015578.001.0001. http://dx.doi.org/10.7551/mitpress/9780262015578.001.0001.

Smolensky, Paul, Richard Thomas McCoy, Roland Fernandez, Matthew Goldrick & Jianfeng Gao. 2022. Neurocompositional computing: From the central paradox of cognition to a new generation of ai systems. *AI Magazine* 43(3). 308322. doi:10.1002/aaai.12065. http://dx.doi.org/10.1002/aaai.12065.

Song, Yuhang, Beren Millidge, Tommaso Salvatori, Thomas Lukasiewicz, Zhenghua Xu & Rafal Bogacz. 2024. Inferring neural activity before plasticity as a foundation for learning beyond backpropagation. *Nature Neuroscience* 27(2). 348–358. doi:10.1038/s41593-023-01514-1. https://www.nature.com/articles/s41593-023-01514-1.

Steinert-Threlkeld, Shane. 2020. An explanation of the veridical uniformity universal. *Journal of Semantics* 37(1). 129–144. doi:10.1093/jos/ffz019. https://academic.oup.com/jos/article/37/1/129/5683663.

Steinert-Threlkeld, Shane. 2021. Quantifiers in natural language: efficient communication and degrees of semantic universals. *Entropy* 23(10). 1335. doi:10.3390/e23101335. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8570335/.

Steinert-Threlkeld, Shane, Nathaniel Imel & Qingxia Guo. 2023. A semantic universal for modality. *Semantics and Pragmatics* 16(1). 120. doi:10.3765/sp.16.1. http://dx.doi.org/10.3765/sp.16.1.

Steinert-Threlkeld, Shane & Jakub Szymanik. 2019. Learnability and semantic universals. *Semantics and Pragmatics* 12(4). 139. doi:10.3765/sp.12.4. http://dx.doi.org/10.3765/sp.12.4.

Steinert-Threlkeld, Shane & Jakub Szymanik. 2020. Ease of learning explains semantic universals. *Cognition* 195. 104076. doi:10.1016/j.cognition.2019.104076. http://dx.doi.org/10.1016/j.cognition.2019.104076.

Strohmaier, David & Simon Wimmer. 2022. Contrafactives and Learnability. *Proceedings of the Amsterdam Colloquium* 305–312. https://platform.openjournals.nl/PAC/article/view/21736.

Strohmaier, David & Simon Wimmer. 2025. Contrafactives, learnability, and production. *Experiments in Linguistic Meaning* 3. 395–410. doi:10.3765/elm.3.5810. https://journals.linguisticsociety.org/proceedings/index.php/ELM/article/view/5810.

Szymanik, Jakub. 2016. *Quantifiers and Cognition: Logical and Computational Perspectives*. Springer International Publishing. doi:10.1007/978-3-319-28749-2. http://dx.doi.org/10.1007/978-3-319-28749-2.

Templeton, Adly, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones et al. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *URL https://transformer-circuits. pub/2024/scaling-monosemanticity* 1.

Uegaki, Wataru. 2023. The informativeness/complexity trade-off in the domain of boolean connectives. *Linguistic Inquiry* 55(1). 174–196. doi:10.1162/ling_a_00461. https://doi.org/10.1162/ling_a_00461.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser & Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30.

Yang, Yuan & Steven T Piantadosi. 2022. One model for the learning of language. *National Academy of Sciences* 119(5). e2021865119. doi:10.1073/pnas.2021865119.

Youn, Hyejin, Logan Sutton, Eric Smith, Cristopher Moore, Jon F. Wilkins, Ian Maddieson, William Croft & Tanmoy Bhattacharya. 2016. On the universal structure of human lexical semantics. *National Academy of Sciences* 113(7). 1766–1771. doi:10.1073/pnas.1520752113. https://www.pnas.org/doi/abs/10.1073/pnas.1520752113. National Academy of Sciences.

Yu, Yong, Xiaosheng Si, Changhua Hu & Jianxun Zhang. 2019. A review of recur-

rent neural networks: LSTM cells and network architectures. *Neural Computation* 31(7). 12351270. doi:10.1162/neco_a_01199. http://dx.doi.org/10.1162/neco_a_01199.

Zaslavsky, Noga, Charles Kemp, Terry Regier & Naftali Tishby. 2018. Efficient compression in color naming and its evolution. *National Academy of Sciences* 115(31). 79377942. doi:10.1073/pnas.1800521115. http://dx.doi.org/10.1073/pnas.1800521115.

Zhu, Zi Long. 2019. *Machine Learning for Semantic Universals*: Universiteit van Amsterdam B.Sc. Informatica Thesis. https://scripties.uba.uva.nl/search?id=record_24958.

Christopher Haberland
Department of Linguistics
University of Washington
Guggenheim Hall 4th Floor
Box 352425
Seattle, WA 98195-2425
haberc@uw.edu

Shane Steinert-Threlkeld
Department of Linguistics
University of Washington
Guggenheim Hall 4th Floor
Box 352425
Seattle, WA 98195-2425
shanest@uw.edu