Semantics & Pragmatics Volume 12, Article 4, 2019
https://doi.org/10.3765/sp.12.4

Learnability and semantic universals*

Shane Steinert-Threlkeld
Department of Linguistics
University of Washington

Jakub Szymanik
Institute for Logic, Language and Computation
Universiteit van Amsterdam

Forthcoming in *Semantics & Pragmatics*.

Abstract One of the great successes of the application of generalized quantifiers to natural language has been the ability to formulate robust semantic universals. When such a universal is attested, the question arises as to the source of the universal. In this paper, we explore the hypothesis that many semantic universals arise because expressions satisfying the universal are easier to learn than those that do not. While the idea that learnability explains universals is not new, explicit accounts of learning that can make good on this hypothesis are few and far between. We propose a model of learning — back-propagation through a recurrent neural network — which can make good on this promise. In particular, we discuss the universals of monotonicity, quantity, and conservativity and perform computational experiments of training such a network to learn to verify quantifiers. Our results are able to explain monotonicity and quantity quite well. We suggest that conservativity may have a different source than the other universals.

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

* We thank audiences at the Cognitive Semantics and Quantities Kick-off Workshop, the Paris-Amsterdam-London Logic Meeting of Young Researchers, the CLIC Seminar Series at the CIMeC Center for Mind/Brain Sciences at the University of Trento, and the Amsterdam Computational Linguistics Seminar. For helpful comments and questions, thanks to Johan van Benthem, Fausto Carcassi, Jakub Dotlačil, Wilker Ferreira Aziz, Nina Gierasimczuk, Kees Hengeveld, Dariusz Kalociński, Meica Magnani, Matt Mandelkern, and Rick Nouwen. Special thanks to Daniel Rothschild for very detailed feedback. The editor, Kai von Fintel, and three anonymous referees for this journal provided valuable feedback that has significantly improved the paper. The computational work was carried out on the Dutch national e-infrastructure with the support of SURF Cooperative. The authors have received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007–2013)/ERC Grant Agreement n. STG 716230 CoSaQ.

1 Introduction

At first glance, the natural languages of the world exhibit tremendous differences amongst themselves. After all, learning a second language as an adult is not an easy task. Yet, early in one's linguistics education, one learns that languages do share tremendous amounts of structure and that the differences can be described, circumscribed, and analyzed. Thus arises one of the central questions in linguistic theory: What is the range of variation in human languages? That is: which out of all of the logically possible languages that humans could speak, do they in fact speak? A limitation on the range of possible variation will be a property that all (or, at least almost all) languages share. Such a property will be a linguistic *universal*.

Universals have been discovered at all levels of linguistic analysis. At the phonological level, all languages have consonants and vowels. More robustly, one can say that all languages have at least one unrounded vowel and at least one back vowel.[1] At the syntactic level, all languages have verbs and nouns.[2] Slightly more controversially, generative grammar as an enterprise can be seen as systematically developing syntactic universals. For example, the basic claim that grammatical rules are structure-dependent[3] is a syntactic universal. At the semantic level, it has been proposed that all languages which have shape adjectives also have color and size adjectives.[4] Closer to the topic of the present paper is the claim that all languages have syntactic constituents (Noun Phrases) whose semantic function is to express generalized quantifiers.[5]

Whenever a universal is attested, it is natural to ask for an explanation of its source. *Why* does the universal hold? While significant differences as to the type of answer to this question arise in the phonological and syntactic domains, many theorists search for *cognitive* explanations of semantic universals. Such an explanation would locate the existence of a semantic universal in a feature of human conceptual apparatus with which semantics must interface.

The present paper develops the hypothesis that semantic universals are to be explained in terms of *learnability*, at least in the domain of quantifiers. We focus on

1 See Hyman 2008 for a thorough discussion of phonological universals, including these two, where they are called "Vocalic Universal #3" and "Vocalic Universal #4" on p. 98.

2 See Croft 1990. Newmeyer (2008) calls for hesitation on positing universals about syntactic categories. Hengeveld et al. (2004) discuss examples of languages that seem not to make the distinction.

3 Chomsky 1965, and many others.

4 See Dixon 1977 for that universal. von Fintel & Matthewson (2008) provide an overview of semantic universals.

5 See Barwise & Cooper 1981 for the source of the universal, Hengeveld et al. 2004 for examples of languages lacking NP quantification, and Bach et al. 1995 for discussion. Note that for our purposes the universal can be formulated as a conditional: if a language has NPs then their semantic function is to express generalized quantifiers over the domain of discourse.

quantifiers simply because this is the area where the largest number of substantial semantic universals have been posited. In developing this hypothesis, we do not claim that learnability will be the only source of semantic universals. For example, communicative need may play a key role in some explanations. Rather, we want to explore how far the learnability hypothesis can be pushed in this domain.

In the literature, there are at least two forms that the learnability argument takes. The first one focuses on the fact that a universal can restrict the hypothesis space for a hypothetical learner of a semantic system.[6] Because a universal greatly shrinks the space of possible quantifier meanings, the learner does not have to explore as much. This makes it is easier[7] to learn these meanings. Seen this way, this learnability argument mirrors at the semantic level Chomsky's poverty of the stimulus argument for universal grammar.[8]

At a certain level, this first argument has to be correct: learning in a smaller hypothesis space will invariably be easier than learning in a larger one. Nevertheless, one should not overstate its conclusions, for two reasons. Firstly, it could be that the benefit to learning from moving to a smaller space is quite negligible. Piantadosi (2013), in a paper where he explores Bayesian learning of quantifiers, puts the point very eloquently:

> Likely, the unrestricted space has many hypotheses which are so implausible, they can be ignored quickly and do not affect learning. The hard part of learning, may be choosing between the plausible competitor meanings, not in weeding out a large space of potential meanings. (p. 22)

Secondly, and more fundamentally, this form of argument can only explain why there are universals at all, but not *which* universals one observes. Any proposed semantic universal has the benefit of decreasing the hypothesis space for the language learner. Because of that, this argument cannot distinguish between competing universals and so cannot explain the exact pattern of universals that are attested. The most that could be gleaned from this line of reasoning would be that one should search for *stronger* universals, since they consist in larger reductions of the hypothesis space and so presumably provide a bigger benefit to learning.

6 Barwise & Cooper (1981), Keenan & Stavi (1986), Szabolcsi (2010) all present a form of this argument.

7 Or, in the most extreme version: possible in the first place.

8 The term is coined in Chomsky 1980, though the argument has appeared in many places in his work. See Pullum & Scholz 2002 for an overview and assessment. Both May (1991) and Partee (1992) follow the argument all the way to the Chomskyan conclusion that the meanings of functional expressions like determiners are innate.

The second form of the learnability argument runs as follows: semantic universals hold because expressions satisfying the universal are *easier* to learn than those that do not.[9] Implicit here is a certain linking hypothesis: meanings that are easier to learn are more likely to be lexicalized. While this paper will not address this hypothesis, it is intuitively plausible: languages have words for meanings that are easier to learn and use compositional methods to express more difficult-to-acquire meanings. As it presently stands, however, this second argument has a major lacuna. For no semantic universals has the argument been fully developed. In order to be more than a suggestion, one cannot simply suggest that expressions satisfying a universal are easier to learn but must actually demonstrate that this is so. This can be seen as a challenge.

CHALLENGE: For the semantic universal(s) of choice, provide a model of learning on which expressions satisfying the universal are easier to learn than others.

The present paper aims to meet the CHALLENGE. In particular, we will focus on three universals in the domain of quantifiers: monotonicity, quantity, and conservativity. We propose a model of quantifier learning by showing how to train a certain kind of recurrent neural network to learn to verify quantified sentences. Computer simulations yield promising initial results. For both the monotonicity and quantity universals, we ran two experiments and found in each that a quantifier satisfying the universal is indeed easier to learn by this model than one that does not. The case of conservativity is more complicated: for reasons to be discussed later, we did not expect our learning model to be sensitive to conservativity. Still, we ran two experiments as a kind of benchmark. As expected, in both experiments, a conservative and non-conservative quantifier are indistinguishable in terms of learnability. Against this backdrop, we argue that this might not be a problem since that universal arguably has a different source.[10]

The paper is structured as follows. Section 2 presents a brief introduction to generalized quantifiers and explains the three universals that we will study. Section 3 presents the model of learning — backpropogation-through-time in a recurrent neural network — that we will apply to quantifiers. In Section 4, we present experiments where we apply the model to each of the universals, with mostly positive results. We provide a general discussion of the results and possible objections in Section 5. In particular, we argue that learnability by a recurrent neural network can be viewed

9 Peters & Westerståhl (2006) (p. 173) allude to this argument for one of the quantifier universals to be discussed later.

10 We note that an earlier body of work applied tools from the learning theory of formal languages to the problem of learning the meanings of quantifiers (Tiede 1999, Gierasimczuk 2007, 2009). The results obtained, however, are too limited in scope to adequately meet the CHALLENGE.

as an operationalization of a general notion of semantic complexity. Finally, we conclude in Section 6 by recording some future directions of research.

2 Quantifier universals

The universals that we focus on have to do with quantifiers, which are the semantic objects expressed by *determiners*. A determiner is an expression taking a common noun as an argument and generating a Noun Phrase. We will assume a division of the determiners in to two classes: simple and complex. Examples of simple determiners are *all, some, no, few, most, five*. Examples of complex determiners are *all but five, fewer than three, at least eight or fewer than five*. Note that we do not at present provide a full account of exactly what the distinction amounts to. For example, while being monomorphemic certainly suffices for being simple, we leave it open that some determiners that are not monomorphemic will still count as simple.[11]

As a first approximation, and following the influential Barwise & Cooper 1981, we assume that determiners denote type $\langle 1,1\rangle$ generalized quantifiers. These can be thought of as being relations between two subsets of a given domain of discourse. For example:

$$\begin{aligned}[\![\text{every}]\!] &= \{\langle M,A,B\rangle :\ A \subseteq B\}\\ [\![\text{at\_most\_3}]\!] &= \{\langle M,A,B\rangle :\ |A\cap B| \leq 3\}\\ [\![\text{most}]\!] &= \{\langle M,A,B\rangle :\ |A\cap B| > |A\setminus B|\}\end{aligned}$$

Before proceeding, a few small notes on terminology. As a shorthand, we will say that a determiner has a certain semantic property to mean that the quantifier that the determiner denotes has that property. Sometimes, for a determiner like *every*, we will write every as a shorthand for $[\![\text{every}]\!]$. We will use Q and its ilk as variables over quantifiers. Because quantifiers are viewed as set-theoretic objects, we will write $\mathscr{M} \in \mathsf{Q}$ when a structure/model $\mathscr{M}$ belongs to a quantifier.[12] In other words, when a sentence *Det N VP* is true when interpreted in a model $\mathscr{M}$, we will write $\langle M, [\![\text{N}]\!], [\![\text{VP}]\!]\rangle \in \mathsf{Det}$. In the remainder of this section, we introduce three prominent semantic universals about quantifiers.

2.1 Monotonicity

To motivate our first universal, consider the following sentences.

11 Arguably, *most* is not monomorphemic. See Hackl 2009, Kotek et al. 2011a,b, Solt 2016. Moreover, some argue that a much wider class, including *no* and *few*, are also not monomorphemic. But these arguably should count as simple for the purpose of formulating semantic universals.

12 See Peters & Westerståhl 2006 for a thorough exposition of quantifiers in this tradition.

(1) a. Many French people smoke cigarettes.
b. Many French people smoke.

It is clear that (1a) entails (1b): the former cannot be true without the latter being true. Similarly, this entailment does not depend on the choice of the restrictor — *French people* — or scopes — *smoke cigarettes* and *smoke* — so long as the latter scope is strictly more general than the former. Moreover, competent speakers of English recognize this fact easily. What speakers thereby implicitly know is that *many* is upward monotone:

(2) Q is *upward monotone* if and only if whenever $\langle M, A, B\rangle \in \mathsf{Q}$ and $B \subseteq B'$, then $\langle M, A, B'\rangle \in \mathsf{Q}$.

By contrast, the pattern seems to reverse if we replace *many* with *few*, as seen in the following examples.

(3) a. Few French people smoke cigarettes.
b. Few French people smoke.

Here, (3b) entails (3a). This is the reverse of the previous case: now, truth is preserved when we move from a more general scope to a more specific scope. In this case, we say that *few* is downward monotone:

(4) Q is *downward monotone* if and only if whenever $\langle M, A, B\rangle \in \mathsf{Q}$ and $B \supseteq B'$, then $\langle M, A, B'\rangle \in \mathsf{Q}$.

Finally, a determiner is *monotone* if and only if it is either upward or downward monotone. The reader can verify that all of the simple determiners mentioned at the beginning of the section are monotone. This appears to be no accident of our choice of English or of that particular list of simple determiners. Barwise & Cooper (1981) proposed the following semantic universal.

MONOTONICITY UNIVERSAL: All simple determiners are monotone.

This universal rules out quantifiers such as an even number of and at least 6 or at most 2: increasing or decreasing the set B can cause the cardinality of $A \cap B$ to change in a way that flips the truth value of sentences with those determiners, so they are not monotone. The claim then is that no simple determiner in any natural language denotes those quantifiers.

2.2 Quantity

Our second universal captures the intuition that determiners express *general* relations between (the denotations of) their restrictor and scope. Whether or not a sentence of

the form *Det A B* should not depend on the identity of any particular A or B, nor on the manner of presentation of those sets. We will build up to the next universal in stages, beginning with an idea borrowed from discussions of logical constants. A *permutation* of a set is a bijection from that set to itself. Permutations can be lifted from sets to models with that set as its domain of discourse in a natural way. We can then say what it is for a quantifier to be logical.

(5) Q is *logical* if and only if for all sets M and permutations $\pi : M \to M$, $\langle M,A,B\rangle \in \mathsf{Q}$ if and only if $\langle \pi(M),\pi(A),\pi(B)\rangle \in \mathsf{Q}$.

In their seminal paper, Keenan & Stavi (1986) propose the universal that "Monomorphemic dets are logical" (p. 311). This rules out expressions such as possessives (e.g. *Susan's*) whose truth does depend on a particular element of the model and so might not be preserved when the elements are permuted. Of course, possessives are not monomorphemic; the universal claims that no monomorphemic determiner could have the same meaning as a possessive like *Susan's*.[13]

A slightly stronger universal than logicality appears to hold. It replaces permutation-invariance with isomorphism-invariance. An *isomorphism* between two models is a bijection between their underlying sets that preserves the additional structure of the model. In the case of models of the form $\langle M,A,B\rangle$, this means that A gets mapped to A' and B to B'.

(6) Q is *isomorphism-invariant* if and only if: if $\langle M,A,B\rangle \cong \langle M',A',B'\rangle$, then $\langle M,A,B\rangle \in \mathsf{Q}$ if and only if $\langle M',A',B'\rangle \in \mathsf{Q}$.[14]

Peters & Westerståhl (2006) formulate and defend the universal that all simple determiners are isomorphism-invariant.[15] In other words, they endorse the following universal.[16]

QUANTITY UNIVERSAL: All simple determiners are isomorphism-invariant.

To understand what quantifiers this universal rules out as the denotations of simple determiners, note the following fact: $\langle M,A,B\rangle \cong \langle M',A',B'\rangle$ if and only if the four sets $A\cap B$, $A\setminus B$, $B\setminus A$, and $M\setminus(A\cup B)$ have the same cardinality as their primed counterparts.[17] So the QUANTITY UNIVERSAL says that the truth value of a

13 See Peters & Westerståhl 2013 for a thorough analysis of possessives.

14 Generalized quantifier theory, as developed in mathematical logic, generally builds isomorphism-invariance into the definition of a quantifier. See Mostowski 1957, Lindström 1966 for the founding documents of that tradition. For application to natural language, however, we do not impose the requirement but see it as an additional constraint on quantifiers.

15 Their exact wording (p. 330): "All lexical quantifier expressions in natural languages denote ISOM quantifiers."

16 The name 'Quantity' comes from van Benthem (1984), who uses it in the present sense.

17 See Peters & Westerståhl 2006, p. 158.

simple sentence of the form *Det N VP* depends only on those four quantities. This rules out lexical items from having the same meaning as exceptive phrases like *all ... except engineers* in sentences such as:

(7) All students except engineers must take a creative writing class.

In particular, the truth of this sentence depends on membership in a fixed set of engineers, and not merely the sizes of the sets built out of the students and those taking a creative writing class.

Additionally ruled out as candidate meanings of simple determiners are quantifiers that depend on the *manner of presentation* of the restrictor and scope. For example, consider the following.

(8) The first three students to solve the problem will get extra credit.

(9) Every other house on that block is vacant.

The truth of both (8) and (9) depends on the *order* in which elements of the restrictor — *students* and *house on that block*, respectively — are inspected. It seems that no language has a lexical item *fthree* which has the meaning of *the first three* in (8).[18] Similarly, one can imagine other sentences whose truth depends on the spatial arrangement of the restrictor. All such expressions are ruled out as possible determiners by the QUANTITY UNIVERSAL.

2.3 Conservativity

Our final universal — arguably the most widely discussed of the three — captures the intuition that the restrictor genuinely restricts what a sentence talks about. That is, sentences of the form *Det N VP* are in some sense about the *N*s and nothing else. That this universal holds can be observed by noting the felt equivalence between the following pairs of sentences.

(10) a. Every student passed.
b. Every student is a student who passed.

(11) a. Most Amsterdammers ride a bicycle to work.
b. Most Amsterdammers are Amsterdammers who ride a bicycle to work.

The formal concept at play here has been called conservativity.

(12) Q is *conservative* if and only if $\langle M, A, B\rangle \in \mathsf{Q}$ if and only if $\langle M, A, A \cap B\rangle \in \mathsf{Q}$.

18 An anonymous referee observes that while this is true, *first* is a widely attested lexeme. While true, we do not think this constitutes a counter-example to the universal because *first* on its own is not a determiner. We find it plausible to analyze *first* in a sentence like *the first house is blue* as an adjective modifying *house*; the resulting NP then combines with the determiner *the*.

Barwise & Cooper (1981) formulated and defended the following universal.[19]

CONSERVATIVITY UNIVERSAL: All simple determiners are conservative.[20]

This universal rules out quantifiers that depend on other portions of the model besides A, such as $B \setminus A$. As an example, there is no determiner *equi* in any language such that the following two sentences are equivalent in meaning.

(13) a. Equi students are at the park.
b. The number of students is the same as the number of people at the park.

This concludes the presentation of the semantic universals to be studied here. Because we primarily focus on the explanation/source of semantic universals, we do not present a detailed defense of each universal. Rather, we assume that the universals hold and attempt to explain why they do in terms of learnability. That being said, we know of virtually no counter-examples to the three universals being studied.[21]

3 The learning model

Recall that our CHALLENGE was to provide a model of learning on which expressions satisfying semantic universals were easier to learn than those that do not. Having now explained three such universals in the domain of quantifiers, we develop model of learning quantifiers. In the next section, we present experiments showing that this model can meet the CHALLENGE.

The basic idea will be to train a *neural network* to learn how to verify and falsify quantified sentences. A neural network is a computational device modeled after the methods of computation and communication in biological nervous systems.[22] Such

19 Because the term *conservative* was not introduced until Keenan & Stavi 1986, the original formulation was in terms of a quantifier *living on* a witness set. We follow the norm of formulating in terms of conservativity for concision.

20 In fact, conservativity often is a claim about all determiners, not just the simple ones. This claim sits well with the view we defend in Section 4.3 that this universal is not a constraint on the lexicon but arises from the workings of the syntax-semantics interface.

21 Most prominently, 'only' and the reverse proportional reading of 'many' (first observed in Westerståhl 1985) have been claimed to be counter-examples to CONSERVATIVITY. One can argue that neither is a determiner: the former is an adverb (von Fintel 1997) and the latter a gradable adjective (Romero 2015). Or one can observe that they are 'conservative on their second argument' and attempt to assimilate this to standard conservativity. See von Fintel & Keenan 2018 for a recent discussion and references.

22 See Rumelhart et al. 1986a,b for an overview of early pioneering work applying neural networks to human cognition. For modern overviews of this area of research, we recommend Nielsen 2015, Goodfellow et al. 2016.

a network consists of a number of *nodes*, which are arranged sequentially in *layers*. Activation — a numerical quantity — travels through the layers because nodes in one layer are connected to those in the subsequent one. The connections between nodes can have different weights, reflecting how important the activation in one node is to another. Such a network looks schematically like Figure 1.

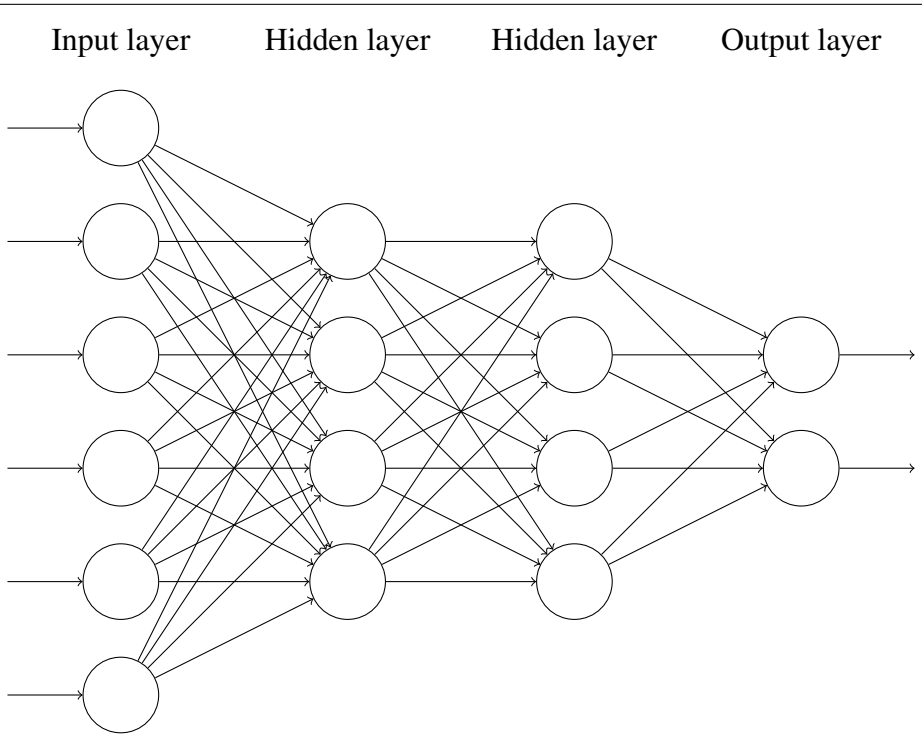

Figure 1 A multi-layer feed-forward neural network.

The first layer is called the *input layer*. The final layer is called the *output layer*. If the input layer has n nodes and the output layer has m nodes, then the network computes a function from $\mathbb{R}^n$ to $\mathbb{R}^m$. The layers — if there are any — in between the input and output layers are called hidden layers. Computation works as follows: each node computes a weighted combination of the activations of the nodes that connect to it and then applies a nonlinearity. Somewhat more concretely, for a non-input layer l, we have that

$$\vec{a}_l = f\left(W^l \cdot \vec{a}_{l-1} + \vec{b}^l\right) \tag{14}$$

where $\vec{a}_l$ is the vector of activations in layer l, W^l is a matrix containing the *weights* of the connections (i.e. W_{ij} is the strength of the connection from node i in layer

$l-1$ to node j in layer l), $\vec{b}^l$ is a vector of biases, and f is some non-linear function applied point-wise.[23]

Such a network learns to approximate a given function by gradually updating the weights and biases in a way that it moves closer to the given function. Formally, this is done by (stochastic) *gradient descent*. Letting $\vec{\theta}$ denote a long vector containing all of the parameters of the network (i.e. all the weights and biases), we can think of the network as computing a function of these parameters and the input, which we will denote $\mathsf{NN}(\vec{\theta},\vec{x})$, where $\vec{x}$ is an input. The learning will be supervised: we have a set of data points $\{\vec{x}^i,\vec{y}^i\}_{i\in I}$, indexed by a finite set I, which (partially) capture the given function's input-output relationship. We assume that there is a total loss function, which is the mean of a 'local' error function ℓ. ℓ is defined on $\mathbb{R}^m\times\mathbb{R}^m$, and measures how close the network's output is to the true output.

$$L=\tfrac{1}{|I|}\textstyle\sum_i \ell(\mathsf{NN}(\vec{\theta},\vec{x}^i),\vec{y}^i) \tag{15}$$

Gradient descent works by considering L as a function of $\vec{\theta}$, and then moving around that space towards lower and lower values of L. Formally, at iteration t of training, the parameters of the network are updated by

$$\vec{\theta}_{t+1}\leftarrow\vec{\theta}_t-\alpha\nabla_{\vec{\theta}}L \tag{16}$$

where α is a learning parameter. The gradient $\nabla_{\vec{\theta}}L$ can be computed by the famous *backpropagation* algorithm. Intuitively, a forward pass through the network generates a guess, after which an error ℓ is calculated. This error can be sent in a backward pass through the network to compute the partial derivative with respect to each parameter.

In practice, more complicated update rules than (16) — with learning rates that are not constant — are deployed. Similarly, stochastic gradient descent improves on this by updating after a *mini-batch* of data points (as small as one example) is processed, instead of only updating after all of the data have been processed. Conceptually, however, the algorithms work the same way: the weights and biases of the network are updated in such a way that loss is reduced, moving the network's function closer to the true function.

To train a neural network to learn the meanings of quantifiers, we will have it learn to do *quantifier sentence verification*. That is, we want the input to the network to be a pair $\langle\mathscr{M},Q\rangle$ of a model and a quantifier expression, and the network will output a 1 or a 0 corresponding to whether $\mathscr{M}\in\mathsf{Q}$ or not. More precisely, the network will output a probability: how strongly the network 'believes' that $\mathscr{M}\in\mathsf{Q}$.

Two features of this task require moving to a model slightly richer than a standard feed-forward neural network as just described. Firstly, the models belonging to a quantifier can come in many different sizes, but these neural networks require a

23 Common choices here are the sigmoid $\sigma(x)=\frac{1}{1+e^{-x}}$ and hypertangent $\tanh(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}}$.

fixed-length input. In practice, one often extracts *features* from an input, so that variable-sized inputs get mapped to a fixed-sized representation. We do not, however, want to pre-select the features of a model that will be relevant to the quantifier verification task, preferring to give the network the raw model. Secondly, to model quantifiers like first three that fail the QUANTITY UNIVERSAL, we want the model to be presented *sequentially* to the network, so that it can be sensitive to the order of presentation of objects.

So-called *recurrent neural networks* overcome both of these limitations. The key innovation in these networks is that they have 'loops': as they process a sequential input, they maintain a state that gets passed on to the next step.[24] Networks of this type are trained by a method called Backpropagation-Through-Time.[25] Essentially, the loops in the network are 'unfolded' for as many steps as in the input sequence, and standard backpropagation is used to calculate the gradients. Figure 2 depicts the architecture and this unraveling schematically. In this Figure and in what follows, we omit the use of arrows to denote vectors in order to improve readability.

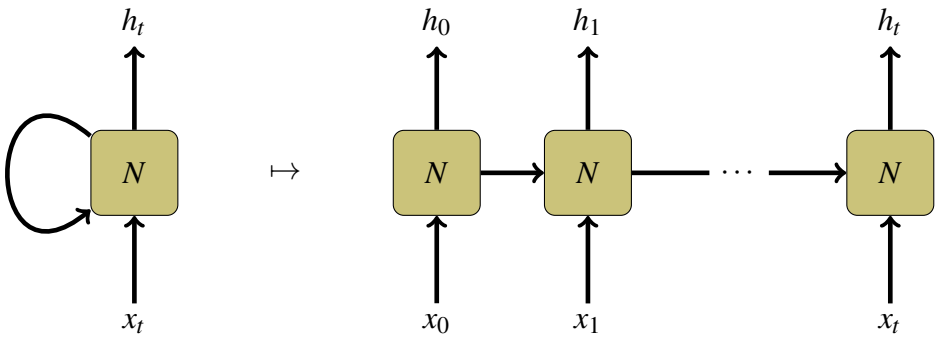

Figure 2 A recurrent neural network (RNN), unrolled as in Backpropagation-Through-Time. The x_i are the input sequence of vectors, and the h_i are output vectors. At time-step i, an RNN receives both x_i and h_{i-1} as inputs. (At the first time-step, some h_0 must be fed into the network. Typically, this will be all zeros or a random vector.) The boxed N represents some mathematical function, usually a kind of neural network.

The particular form of RNN that we will use is called a *long short-term memory (LSTM)* network.[26] While these were introduced to solve a technical problem in

24 For early applications, see Jordan 1986, Elman 1990, Bengio 1991.
25 See Werbos 1988 for an early version of the algorithm.
26 Introduced in Hochreiter & Schmidhuber 1997.

training RNNs,[27] they also admit of an intuitive interpretation. As the network processes a sequence, it maintains a state c_t. At item t in the sequence, the network chooses which bits of the cell to forget, and which bits of the input to write to the cell. In this way, the network maintains a form of memory as it processes a sequence. Mathematically, the network computes the following function, depicted schematically in Figure 3, and subsequently explained intuitively.

(17) LSTM computation:

$$
\begin{aligned}
f_t &= \sigma\left(W^f \cdot h_{t-1}x_t + b^f\right) \\
i_t &= \sigma\left(W^i \cdot h_{t-1}x_t + b^i\right) \\
\hat{c}_t &= \tanh\left(W^c \cdot h_{t-1}x_t + b^c\right) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \hat{c}_t \\
o_t &= \sigma\left(W^o \cdot h_{t-1}x_t + b^o\right) \\
h_t &= o_t \odot \tanh\left(c_t\right)
\end{aligned}
$$

In the equations above, $\odot$ denotes component-wise multiplication, and $h_{t-1}x_t$ represents vector concatenation. Note that the computation of $f, i, \hat{c}, o$ are instances of the basic neural network layer activation computation from equation (14), with $h_{t-1}x_t$ as the 'previous activation'.

The equations f_t and i_t should be thought of as forget and input gates, respectively. Consider f_t. It will be a vector of the same number of dimensions as c_t. The sigmoid activation function used outputs values between 0 and 1. The value of f_t in dimension j can be thought of as how strongly to 'remember' element j of the state vector c_{t-1}. This is due to the element-wise multiplication of f_t and c_{t-1} in the calculation of c_t. For instance, if element j of f_t is 0, then the corresponding element of c_{t-1} will be entirely erased. The input gate i_t works similarly, though it interacts not directly with c_{t-1}, but with a set of 'candidate' values $\hat{c}_t$ for the next state vector c_t. Finally, the equation for c_t encapsulates the idea that values of the state are forgotten according to f_t and new ones are written according to i_t. One last output gate o_t filters which components of the cell c_t to output at each time step.

Before proceeding, we make two remarks motivating our choice of LSTM networks. Firstly, these networks have become the gold standard type of neural network for processing sequential data. They have been crucial components in models that have achieved remarkable results in tasks like language modeling,[28] image and

27 The so-called problem of vanishing and exploding gradients: the unrolling used in BPTT tends to produce gradients that either become very close to zero or very large.

28 Graves 2013, Jozefowicz et al. 2016

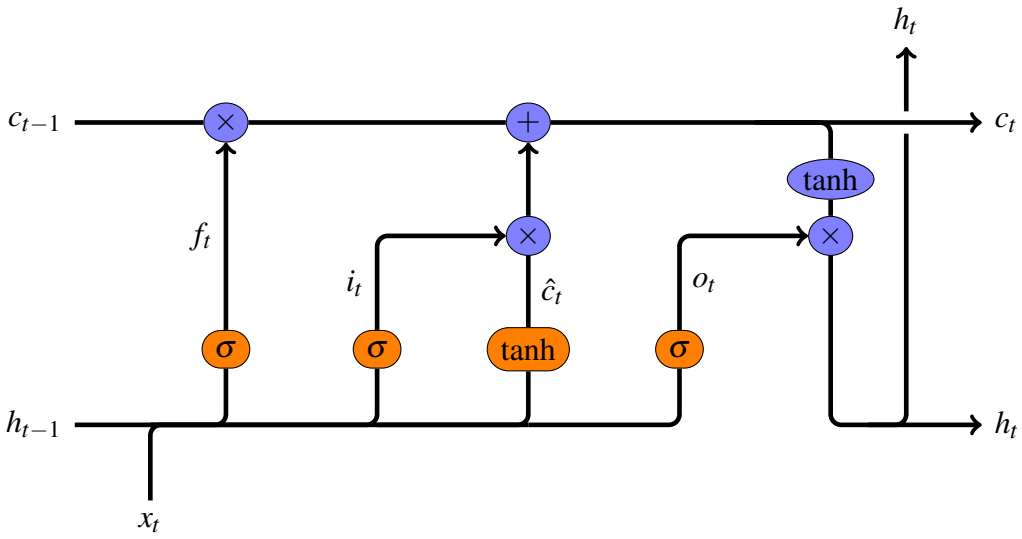

Figure 3 A long short-term memory (LSTM) network. The orange nodes represent neural network layers: matrix multiplication by a weight matrix (plus addition of bias) before a point-wise nonlinearity as labeled. The blue nodes represent pointwise application of a function. The merging of two arrows represents vector concatenation and the splitting of an arrow represents copying.

video captioning,[29] and machine translation.[30] To that end, we have not tailor-made a network architecture to our task, but grabbed one off-the-shelf and applied it to the task of learning quantifiers. Secondly, recent work in neuroscience shows that gating mechanisms not unlike those that regulate the flow of information in an LSTM (e.g. the forget and input gates) underly the operation of working memory in humans.[31] These two factors make our choice of network model extremely natural and well-suited to the task of addressing the CHALLENGE: the model appears to be domain-general and biologically plausible.

Here is how we will apply an LSTM to the task of verifying a quantifier. We want its input to be a sequence, representing a model, together with a quantifier, and its output to be a guess at the truth-value. Thus, it is a *sequence classification* task.[32] The relevant truth-value to be guessed is for a sentence of the form 'Q A

29 Xu et al. 2015, Donahue et al. 2017

30 Sutskever et al. 2014, Wu et al. 2016

31 McNab & Klingberg 2008, Gisiger & Boukadoum 2011

32 One may observe that our present model thus generalizes the semantic automata approach to quantifiers, pioneered by van Benthem (1986) (see Steinert-Threlkeld & Icard III. 2013, Steinert-Threlkeld

B'. Because our focus is on learning the meaning of the quantifier and not on any syntactic parsing, we take the A and B to be schematic and present the model with objects labeled for their membership in those sets.

The zones of a model — $A \cap B, A \setminus B, B \setminus A, M \setminus (A \cup B)$ — will be encoded as one-hot vectors. For example, an element of $A \cap B$ will be encoded as $\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}$. We assume the models are enumerated, and run through the enumeration, generating a sequence of vectors. Finally, each quantifier is also labeled with a one-hot vector, in as many dimensions as there are quantifiers. The vector for the quantifier is concatenated to each element's vector. One can think of this in the following way: as the network processes the model sequentially, it always has access to the sentence that it's attempting to verify in much the same way as human participants often do in sentence-verification experiments. This should make the learning task slightly easier, since the network does not have to remember what quantifier it is verifying. In (18), we provide a detailed example of this encoding.

(18) Encoding a model as a sequence for the LSTM.

	$\in A$?	$\in B$?	x_i
o_1	✓	✓	$\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 1\end{bmatrix}$
o_2	✓	×	$\begin{bmatrix}0 & 1 & 0 & 0 & 0 & 1\end{bmatrix}$
o_3	×	✓	$\begin{bmatrix}0 & 0 & 1 & 0 & 0 & 1\end{bmatrix}$
o_4	✓	✓	$\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 1\end{bmatrix}$
o_5	×	×	$\begin{bmatrix}0 & 0 & 0 & 1 & 0 & 1\end{bmatrix}$

In the above table, the o_i are the five objects of a model, presented in that order. The next two columns indicate whether each object belongs to the set A (the restrictor) or B (the nuclear scope), respectively. Finally, the x_i column represents the input at step i to the LSTM for this example.
In this example, we assume that the network is being trained on two quantifiers — every and some — and that they are 'ordered' in that way. The final two dimensions of each x_i encode that the desired/intended output is the truth value of 'some A is a B'. When it is being asked to output the truth value of 'every A is a B', the final two dimensions will be $\begin{bmatrix}1 & 0\end{bmatrix}$ for each x_i.
Finally, the true label y for this example will be $\begin{bmatrix}1 & 0\end{bmatrix}$, because the sentence is indeed True. The truth-value False is represented by the vector $\begin{bmatrix}0 & 1\end{bmatrix}$.

2016, Szymanik 2016 for later work in the tradition). There, different kinds of automata process sequences of letters from an alphabet in much the same way that our LSTM will process a sequence of vectors. A primary advantage of using recurrent neural networks for modeling quantifiers is the ability to apply back-propagation as a model of learning.

After the LSTM processes the sequence corresponding to a model and quantifier, the final output is passed to a one-layer feed-forward neural network with two outputs, corresponding to True and False. This output layer has a softmax activation function, so that the resulting activations are probabilities.[33] Full details of the network architecture and the data generation process will be described in the following section.

This choice of input captures some necessary features for addressing the CHALLENGE. If, for instance, we encoded models by the cardinalities of the four respective sets, we would not be able to represent quantifiers that do not satisfy QUANTITY. Similarly, if we did not represent all four sets but only the sets $A \cap B$ and $A \setminus B$, we would not be able to represent non-conservative quantifiers.[34]

To complete the description of the learning model, we must specify what loss function we will be minimizing. In tasks like ours where the output is a probability distribution, the standard choice is *cross-entropy*. This function can be seen as capturing the 'distance' between two probability distributions. Or, since it's not symmetric, the amount of 'work' that one would have to do to transform a given distribution into a target one. For discrete distributions, the general form is:

$$E(p,q) = -\sum_{i=1}^{m} p(x_i) \ln q(x_i) \tag{19}$$

In our case, this takes on a particularly simple form. This is because the target distribution p comes from our training data, and so assigns all of the probability to the correct truth-value, and none elsewhere. Thus, for $y \in \{0,1\}$ being the correct truth-value, our local error function will be:

$$\ell(\mathsf{NN}(\theta,x),y) = -\ln(\mathsf{NN}(\theta,x)_y) \tag{20}$$

This makes good intuitive sense. Because y is the correct truth-value, $\mathsf{NN}(\theta,x)_y$ is the probability that the network assigns to the correct truth-value. When this probability equals 1 (i.e. when the network completely correctly guesses the right truth-value), then ℓ is 0. And as this probability gets farther and farther away from 1, the loss increases. Plugging this local error function into a gradient descent algorithm, then, means that the network will learn to assign higher and higher probability to the correct truth-value.

33 With v a vector, $\text{softmax}(v)_i = e^{v_i} / \sum_j e^{v_j}$. We could use just one node, since the guess is a probability, but having two outputs allows for easier generalization to other classification tasks.

34 We note that our choice of input, as motivated by the CHALLENGE, explains some differences with recent approaches to learning quantifiers with neural networks (Sorodoc et al. 2016, Pezzelle et al. 2017, Sorodoc et al. 2018). They focus primarily on the ability of networks to learn quantifiers from images (possibly augmented with text). We use a more austere input to factor out tasks like image processing and syntactic parsing from the purely semantic learning that we are interested in. Interestingly, Sorodoc et al. (2018) find that a neural image- and language-processing architecture which first processes the restrictor before combining it with the nuclear scope achieves superior performance.

4 Experiments

We are now in a position to directly address the CHALLENGE: we have three proposed semantic universals about quantifiers and a model of learning quantifiers. We can thus ask, for each of the universals, the following question: are expressions satisfying the universal easier to learn (by an LSTM) than those that do not? In this section, we present three experiments, one for each universal. Because all experiments shared the same methodology, we first describe the methods. All code for running the simulations and the data generated are available at http://github.com/shanest/quantifier-rnn-learning.

For each universal, we do the following: choose a pair of quantifiers, one satisfying the universal and one not satisfying it. We then run some number of trials of training an LSTM to learn those two quantifiers. Multiple trials are needed for a robustness check since the learning is a stochastic process.[35] For each trial, we measure how long it took the network to converge for each quantifier, and compare the two, where convergence means having reached and maintained a suitably high threshold of accuracy.[36] To meet the CHALLENGE, we hope that the networks systematically converge earlier for the quantifier satisfying the universal.

More concretely, Algorithm 1 below depicts our data generation algorithm. Essentially, a quantifier is drawn at random, and a sequence corresponding to a model of a randomly-chosen size is also generated. We then add the corresponding data point to the data set, avoiding duplicates. Finally, we shuffle the data, and balance it so that every quantifier/truth-value pair has the same number of data points. The balancing is done by under-sampling to the smallest class.[37] We then split the generated data into a training set and a test set. In all of our experiments, the split was 70%/30%. The algorithm has three parameters: the maximum length of a model, the number of data points to generate, and a set of quantifiers. For all experiments, the maximum length was 20. The other two parameters varied by experiment and so will be reported for each. We varied the total number of data points generated for the following reason: at the end, the data is balanced so that each quantifier/truth-value pair has the same number of data points, so that the network does not simply learn a bias in the data. We performed this balancing by under-sampling, so that each pair ended up with the same number of data points as the least-frequent quantifier/truth-value pair. Because different quantifiers have a different distribution of truth-values across the space of models (for example, most is true roughly half the time, while all is very rarely true), we varied the total number of

35 Initialization of the LSTM state and of the weights are random, as is the data generation algorithm, both in exactly which input sequences get generated and the order they are presented to the network.

36 We state our precise measure of convergence below.

37 For motivations and methods on balancing data, see He & Garcia 2009.

Algorithm 1 Data Generation Algorithm

Inputs: `max_len`, `num_data`, `quants`

```
data ← []
while len(data) < num_data do
    Choose N uniformly at random from between 1 and max_len
    Choose Q uniformly at random from quants
    cur_seq ← N randomly chosen items from {A∩B, A\B, B\A, M\(A∪B)}
    if ⟨Q, cur_seq⟩ ∉ data then
        Add ⟨Q, cur_seq, cur_seq ∈ Q?⟩ to data
    end if
end while
shuffle(data)
balance_by_undersampling(data)
return train_split(data), test_split(data)
```

data points generated so that the resulting number of data points per quantifier/truth-value pair were roughly the same across experiments once the under-sampling was performed.

For each experiment, we ran 30 trials of learning. Our networks consisted of two stacked LSTM cells, each with a hidden state of 12 nodes. We stopped the model when the total loss was below 0.01, total mean accuracy for 100 training mini-batches was over 99%, or 4 epochs have passed,[38] whichever came first. We used mini-batches of size 8.[39] We used the Adam optimizer[40] with learning rate 10^{-5}. All of this was implemented using TensorFlow.[41]

For analysis, we measured the *convergence point* for each quantifier: this is the first time step where both the accuracy and the mean accuracy on the *test* set from then until the end of the trial were above 95%. To see whether one quantifier systematically converged earlier than the other, we calculated a paired t-test on the convergence points, which is equivalent to a 1-sample t-test on the differences between the two points. This was chosen because within-trial difference of convergence

38 In machine learning, an epoch is one pass through the entire set of training data. See Nielsen 2015, ch. 1, for this and other terminology.

39 A mini-batch is how much data is used before updating the network. In standard gradient descent, the batch is the size of the entire training data, i.e. the network only computes gradients and updates after seeing all its data. Because this is costly, *stochastic gradient descent* uses smaller batches. In general, the smaller the batch-size, the worse the approximation to the total gradient, so the more chaotic learning will be.

40 Kingma & Ba 2015

41 See http://tensorflow.org and Abadi et al. 2016.

points is more meaningful than comparisons across trials due to the stochastic nature of the learning process.

4.1 Monotonicity

Our first experiment tested the MONOTONICITY UNIVERSAL. Because a quantifier can be either upward- or downward-monotone, we ran one experiment with an upward monotone quantifier and another one with a downward monotone one. In both cases, we generated 100000 data points.

Experiment 1(a) compared at least 4 — an upward-monotone quantifier, meaning $|A \cap B| \geq 4$ — with the quantifier at least 6 or at most 2 — a non-monotone quantifier meaning $|A \cap B| \geq 6$ or $|A \cap B| \leq 2$. The learning curves for all of the 30 trials are plotted in Figure 4. Qualitatively, it appears that at least 4 regularly converges faster than at least 6 or at most 2. The statistics confirm this appearance. A paired t-test of the convergence points found that at least 4 did converge statistically significantly earlier across trials ($t = -9.356$, $p = 2.926 \times 10^{-10}$).

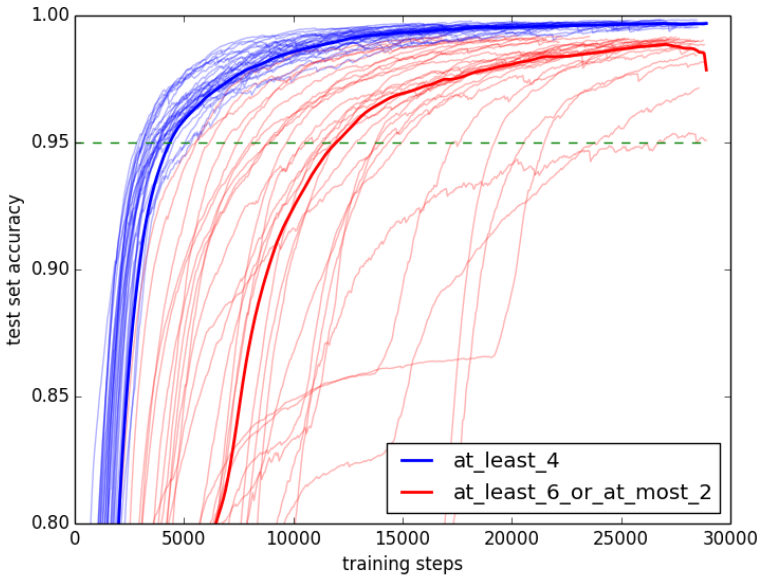

Figure 4 Experiment 1(a) learning curves. The median at each step is in bold.

Experiment 1(b) compared at most 3 — a downward-monotone quantifier, meaning $|A \cap B| \leq 3$ — with the quantifier at least 6 or at most 2. The learning curves for all 30 trials are plotted in Figure 5. Qualitatively, it appears that at most 3 regularly

converges faster than at least 6 or at most 2. The statistics confirm this appearance. A paired t-test of the convergence points found that at most 3 did converge statistically significantly earlier across trials ($t = -15.253$, $p = 2.182 \times 10^{-15}$).

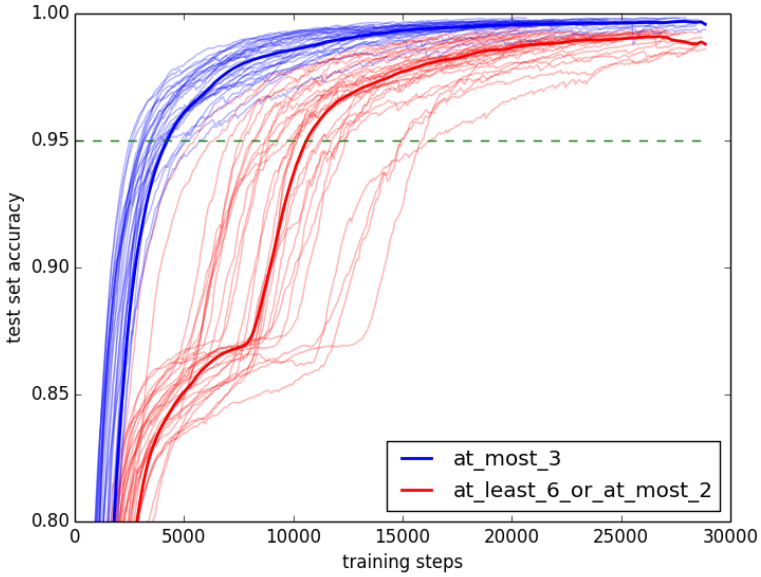

Figure 5 Experiment 1(b) learning curves. The median at each step is in bold.

These results are very encouraging. Both upward- and downward monotone quantifiers are learned significantly more quickly than a corresponding non-monotone quantifier by an LSTM network. In the context of the present study, this supports the argument that the MONOTONICITY UNIVERSAL holds because monotone quantifiers are easier to learn.

4.2 Quantity

Our second experiment tested the QUANTITY UNIVERSAL. First, we compared at least 3 — a quantifier that is quantitative — with first 3 — a quantifier that is not quantitative because it depends on the order in which the restrictor is presented. In this case, we generated 200000 data points. We threw out one trial which failed to reach high enough accuracy for both quantifiers after 4 epochs.[42] The learning curves for the remaining 29 trials are plotted in Figure 6. Qualitatively, while the

42 This trial was close to reaching 95% test set accuracy for each quantifier, so the network was still learning.

separation does not look as strong as in Experiments 1(a) and 1(b), it does appear that at least 3 converges faster than first 3. The statistics confirm this appearance. A paired t-test of the convergence points found that at least 3 did converge statistically significantly earlier across trials ($t = -7.549$, $p = 4.032 \times 10^{-8}$).

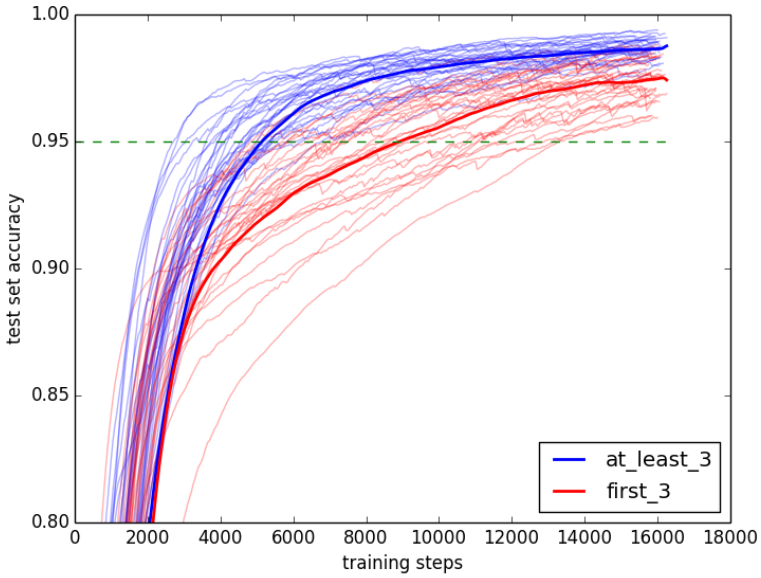

Figure 6 Experiment 2(a) learning curves. The median at each step is in bold.

To test the robustness of this result and ensure that it does not reflect a defect in our learning model, we ran a second experiment to test QUANTITY.[43] In particular, despite having 'memory' in their name, it is known that LSTM networks can in fact have trouble maintaining memory during the course of processing long sequences.[44] So it could be what drove the result in Experiment 2(a) was not a general feature of the learning model, but rather a difficulty in maintaining a memory of the early part of the sequence, to which first 3 is sensitive. Because of this, we ran a second experiment using last 3 instead of first 3: this quantifier exhibits the same order-dependency as first 3 but places less demands on the model's memory of early parts of a long sequence. The learning curves are plotted in Figure 7. Qualitatively, the pattern continues to hold and in fact may be stronger: at least 3 appears to converge much faster than last 3. The statistics confirm this appearance. A paired t-test of the

43 We are grateful to an anonymous referee for suggesting this confound and subsequent experiment.

44 This, for instance, partially explains the significant performance boost from reversing the source sentence in LSTM-based neural machine translation in Sutskever et al. 2014.

convergence points[45] found that at least 3 did converge statistically significantly earlier across trials ($t = -26.453, p = 7.459 \times 10^{-22}$).

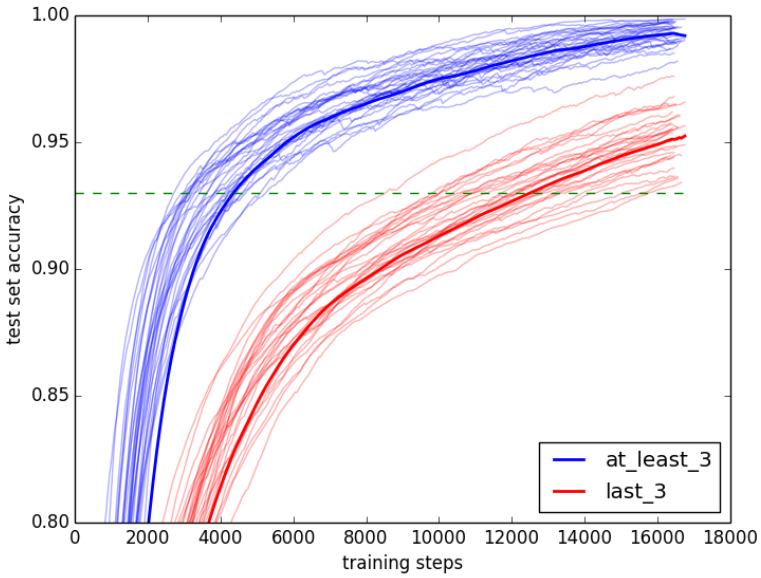

Figure 7 Experiment 2(b) learning curves. The median at each step is in bold.

These results, like those before, are encouraging. We have now seen a second universal — the QUANTITY UNIVERSAL — where a quantifier satisfying the universal is learned more easily by an LSTM network than one that does not. That this pattern has been observed for two very prominent universals also lends support to the general argument of which the CHALLENGE is a missing piece: in general, a universal may hold because expressions satisfying it are easier to learn and such expressions are more likely to be lexicalized.

4.3 Conservativity

Our third and final set of experiments focused on the CONSERVATIVITY UNIVERSAL. Before presenting the results, we note that CONSERVATIVITY appears somewhat different from the previous two universals. While the former two impose robust patterns on the distribution of truth-values for quantified sentences across the space of models, the present universal simply says that one 'zone' of a model — namely,

45 For this experiment, we lowered the threshold to 93%, since many trials did not quite reach 95% accuracy for last 3.

$B \setminus A$ — is irrelevant. In terms of our learning model, conservativity simply entails that one of the four symbols in the input alphabet will never effect the truth-value (i.e. the classification label); but the patterns of the remaining symbols in the sequences can be of any kind. Because of this difference, there exists *prima facie* reason to doubt that the model will distinguish conservative from non-conservative quantifiers. We now present two experiments as a kind of 'sanity check', exhibiting that this is in fact the case. After so doing, we conclude with a brief discussion of ways the model can be extended to possibly tease apart conservative and non-conservative quantifiers and a positive proposal about the different source of this universal.

First, we compared not all — a conservative quantifier meaning that $A \not\subseteq B$ — with not only — a non-conservative quantifier meaning that $B \not\subseteq A$. We chose these two quantifiers following Hunter & Lidz (2013), who taught children to learn a new determiner, *gleeb* or *gleeb'*. The former — *gleeb* — meant not all, while the latter meant not only. They found that children learned the meaning of *gleeb* faster than that of *gleeb'*, suggesting that conservative quantifiers are easier for children to learn.

In this experiment, we gathered 300000 data points. The learning curves for all of the 30 trials are plotted in Figure 8. Qualitatively, there does not appear to be any significant separation in the learning curves for the two quantifiers. The statistics confirm this appearance. A paired t-test of the convergence points found that neither quantifier converged significantly earlier than the other ($t = 1.098$, $p = 0.281$).

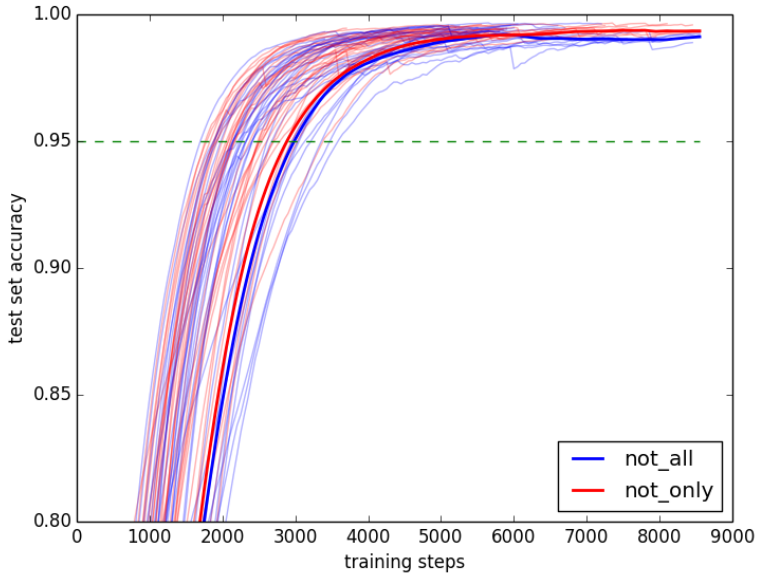

Figure 8 Experiment 3(a) learning curves. The median at each step is in bold.

As expected, not all and not only were learned equally quickly. The *prima facie* intuition in the above can be made more precise in this case: the former says that $A \not\subseteq B$, i.e. that some $A \setminus B$ appears in the sequence; the latter says that $B \not\subseteq A$, i.e. that some $B \setminus A$ appears in the sequence. Because $A \setminus B$ and $B \setminus A$ are simply different labels in our sequences, the two quantifiers are equivalent at a certain level of generality. The network needs to recognize only one 'pattern' — non-emptiness — and learn which character it applies to for which quantifier.

We ran a second experiment, with two quantifiers that are not quite so intimately related. We compared most — with the meaning $|A \cap B| > |A \setminus B|$ — to an invented non-conservative quantifier M, with the meaning $|A| > |B|$.[46] The learning curves for all 30 trials are plotted in Figure 9. Qualitatively, there does not appear to be any significant separation in the learning curves for the two quantifiers. The statistics confirm this appearance. A paired t-test of the convergence points found that neither quantifier converged significantly earlier than the other ($t = 0.762, p = 0.452$).

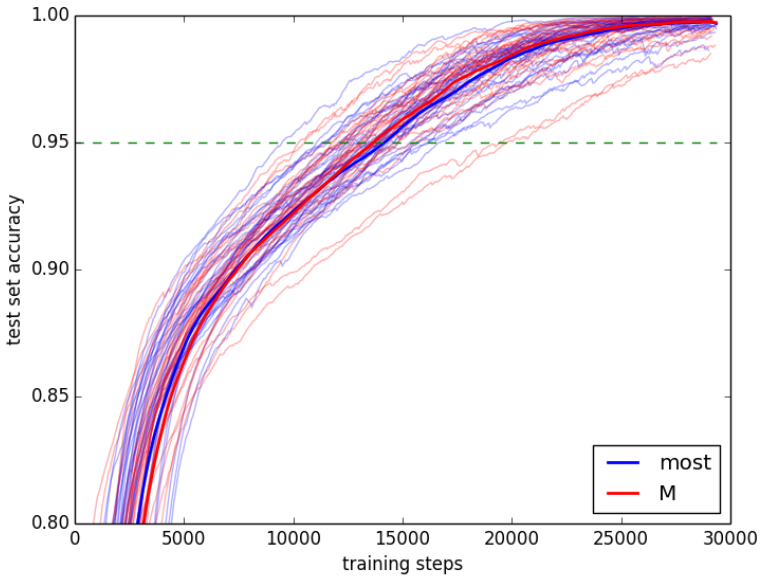

Figure 9 Experiment 3(b) learning curves. The median at each step is in bold.

These results again pass our sanity check: the model cannot distinguish a conservative from a non-conservative quantifier. The situation is partially analogous to the first experiment: because $|A| > |B|$ is equivalent to $|A \setminus B| > |B \setminus A|$, the network

46 See Keenan & Westerståhl 1997, from whence the name M comes. We thank an anonymous referee for suggesting that we test these two quantifiers.

again needs to only learn one pattern, but attach different pairs of labels to it for most and M. The situation is, however, also partially dis-analogous: because both quantifiers depend on $A \setminus B$ but only the latter depends on $B \setminus A$, the result could have been different. That these two patterned with not all and not only is then a welcome null result, strengthening the robustness of our sanity check.

In total then, our model at present cannot meet the CHALLENGE when it comes to the CONSERVATIVITY UNIVERSAL. That being said, there are ways of enriching the present setup in ways that may help. In particular, our minimal pair methodology may be a limiting factor. In particular, since neural networks are known to learn to reflect biases in the data that they are trained on,[47] it is possible that biasing the data by including more conservative quantifiers than non-conservative ones could make the former easier to learn than the latter. We leave this possibility to future work.

While it thus remains possible that the present model could be enriched so as to make conservative quantifiers easier to learn, we contend that the *prima facie* argument at the beginning of this section and our subsequent sanity checks may point to something deeper than a limitation in the current learning model: they could indicate that the source of the CONSERVATIVITY UNIVERSAL differs from that of the other two universals. In particular, a growing number of researchers argue that conservativity is not a constraint on which quantifiers are lexicalized as determiners but is rather an artifact of the syntax-semantics interface.[48] For this reason, many authors develop a so-called *structural* explanation of conservativity. While the details of the proposals need not concern us here, the key idea can be explained: while determiners in principle could denote non-conservative quantifiers, the way that the syntax-semantics interface constructs sentence meanings[49] renders any sentence with a non-conservative determiner truth-conditionally equivalent to a sentence with a conservative determiner.[50] Although nothing in the present paper constitutes an argument for a structural account of conservativity, our null results fit very nicely with such an account: if conservativity ultimately arises as a product of the syntax-semantics interface, it will not be a constraint on the lexicon and so we should not expect its source to be semantic learnability at all.[51] This universal

47 See, among others, Bolukbasi et al. 2016, Buolamwini & Gebru 2018.

48 See Fox 2002, Sportiche 2005, Romoli 2015 for accounts of this type.

49 The key ingredient here is the copy theory of movement.

50 Strictly speaking, such sentences might also be trivial. For this reason, Romoli (2015) assumes that trivial meanings are blocked.

51 We also should not expect a structural account to be sensitive to the distinction between simple and complex determiners. As noted earlier, this universal often does get stated in terms of all determiners, reflecting that it may not be a constraint on the lexicon.

would then fall outside the domain of the CHALLENGE, and so the inability of our model to meet it should not be surprising.[52]

5 Discussion

In total, our results go a long way in meeting the CHALLENGE: the quantifiers in our experiments that satisfy monotonicity and quantity are easier to learn in our model than those that do not. Moreover, although a conservative quantifier is not easier to learn than a non-conservative one, we argued that there are independent reasons to expect the source of that universal to be different than the first two. In this discussion section, we clarify the nature of our argument by discussing three possible objections: that we study non-lexicalized quantifiers, that a notion of semantic complexity really drives the results, and that our learning model is unrealistic.

First, one may feel uneasy that in some of our experiments, the quantifier satisfying the proposed universal appears not to be lexicalized in any language. This holds for not all and at most 3.[53] First, we note that even though no specific quantifier at most n is lexicalized, cardinal *few* is contextually equivalent to some such quantifier. More importantly, however, our argument only delineates the class of quantifiers that are candidates for lexicalization. That is to say, we intend to put upper bounds on what quantifiers simple determiners can denote, but not to exactly demarcate the set of quantifiers denoted in every language. Because exactly which quantifiers satisfying a given universal are denoted by simple determiners varies across language,[54] our goal has been to show merely that quantifiers satisfying the universal are easier to learn and so are better targets for lexicalization.

Second, one might argue that what really explains the universals is a notion of 'semantic complexity', together with the thesis that simple expressions tend to denote less complex meanings. At an intuitive level, this seems to make the same predictions as our results. at least 6 or at most 2, being a disjunction of an upward and a downward monotone quantifier, is more complex than both at least 4 and at most 3. Similarly, first 3, which requires checking that no $A \setminus B$ is observed before three $A \cap B$ are, can be argued to be more complex than at least 3, which only needs to look at $A \cap B$. And, as discussed in the preceding section, not all and not only appear

52 We note that it remains unsolved how to account for 'only' and reverse-proportional 'many' on this structural account of conservativity. Perhaps this provides more reason for arguing that those are not in fact determiners. See footnote 21.

53 If one thinks bare numerals only have an 'exactly' interpretation, then at least 4 would also not be lexicalized. Because, however, it's common to have bare numerals denote at least n and to derive the exactly n interpretation pragmatically, we consider it a candidate to be lexicalized.

54 See Keenan & Paperno 2012 and Paperno & Keenan 2017 for the state-of-the-art knowledge on cross-linguistic patterns in quantifiers. Katsos et al. 2016 is a study of quantifier acquisition cross-linguistically.

to be of equal complexity. Perhaps, then, complexity is the fundamental notion that explains the universals, either by itself explaining learnability or independently.

We present two responses to this line of thought, both hinging on the importance of moving beyond an intuitive notion of semantic complexity to a precise and robust one. On the one hand, because no such notion has been developed and finding one will be difficult, one can view learnability by a recurrent neural network as a kind of operationalization of semantic complexity. That is to say: the best notion of complexity that we have on hand right now just is our notion of learnability. So until an independent and robust general definition of semantic complexity appears, it will be hard to tease apart whether complexity or learnability is more fundamental in the explanation of semantic universals.

On the other hand, there will be difficulties in developing such a notion. First, while the application of tools from computational complexity theory to the semantics of quantifiers have motivated plausible cognitive models,[55] these tools do not make sufficiently fine-grained distinctions to explain the universals we are interested in. So consider, again, the intuitive explanation for why our monotone quantifiers are simpler than the non-monotone ones: the latter are *disjunctive*, while the former are not. And certainly disjunctive quantifiers will be more complex than non-disjunctive ones. This suggests that something like description length in a mental representation language captures semantic complexity.[56] But, as has been known for a long time, what counts as simple according to measures like this depends on what the primitives are.[57] As an example, consider the non-monotone quantifier `exactly n`. This is equivalent to `at least n and at most n`, and so could be argued to be more complex. But that's only if the latter two quantifiers are more primitive. After all, `at least n` is equivalent to `exactly n or more than n`. None of these considerations entail that no notion of semantic complexity can be given; rather, they point to difficulties that will need to be overcome if the intuition is to be made precise. We would welcome a proposal that does overcome these difficulties.

Finally, one can worry that our model of learning does not resemble the sort of learning that children do when learning the meanings of expressions in natural language. The most fundamental worry here concerns the fact that children tend to learn from positive examples, whereas our model requires an even balance of

55 See, e.g., McMillan et al. 2005, Szymanik & Zajenkowski 2010. Szymanik (2016) presents and discusses much of this work.

56 See Tenenbaum et al. 2011 for an overview. Piantadosi (2013) applies the framework to quantifier learning, though with different motivations. Kemp & Regier (2012) uses description length in grammars to explain universal properties of kinship systems in language. Feldman (2000) shows that Boolean complexity of binary feature concepts correlates with their ease of learning.

57 See, for instance, the New Riddle of Induction in Goodman 1955, where it is noted that 'green' and 'blue' become definable if 'grue' and 'bleen' are taken as primitives. The paper Piantadosi et al. 2016 is an attempt to explain exactly what the primitives are in language-of-thought models.

positive and negative examples.[58] It is true that, *ceteris paribus*, one would like a model of learning as close to what is known about the acquisition of quantifiers as possible. Minimally, however, the work presented here stands as a proof of concept: our CHALLENGE was to provide *a* model of learning on which quantifiers satisfying universals are easier to learn than those that do not. We have succeeded on that front and, to that end, demonstrated that the CHALLENGE is not in principle unsolvable. Furthermore, while the exact details of our model may diverge from the best models of acquisition, we do find it plausible that there will be 'cross-model' transfer: the features that make, for instance, monotone quantifiers easier to learn for our model than non-monotone quantifiers are likely to make them easier to learn for other models as well.

6 Conclusion

Let us take stock. In this paper, we have been developing a particular kind of answer to the question about the origin of semantic universals. According to this answer, such universals arise because expressions satisfying the universal are easier to learn than those that do not. This results in a CHALLENGE: develop a model of learning on which the former claim holds. In this paper, we have done just that. In particular, we have shown how to train a long short-term memory neural network to learn to verify quantified sentences and explored three semantic universals for quantifiers: monotonicity, quantity, and conservativity. For the first two universals, our model adeptly meets the challenge: monotone and quantitative quantifiers are learned faster than those that do not. This does not hold true for conservativity; but there are independently motivated arguments to suggest that conservativity has a different source than the other universals.

While these results constitute a promising answer to the CHALLENGE, future work can extend in several different directions. Firstly, we can conduct more and larger experiments. For example, instead of our minimal pair methodology, one would like to train a *single* network to learn a significantly wider range of quantifiers, using the semantic properties of the quantifiers as predictors for the rate of learning. Technical limitations prevent this approach currently. Secondly, one would like to develop tools to 'look inside' the black box of our trained networks and see how they actually operate. For example, is there a sense in which they learn to verify quantifiers in a way analogous to semantic automata,[59] which are computational devices for verifying quantifiers? Thirdly, similar experiments could be run for other semantic

58 Note that here a positive example is not explicitly a grammatical sentence, but a pairing of a sentence with a scenario in which it is true.

59 See the references in footnote 32.

universals, both within the quantifier domain and in other linguistic domains.[60] As an example, *convexity* of denotations for nouns and adjectives seems robust and mirrors monotonicity for quantifiers.[61] More concretely, see Steinert-Threlkeld 2018 for a similar approach to explaining a semantic universal about responsive verbs. Finally, recall that the general structure of our learnability argument depends on a linking hypothesis: expressions that are easier to learn are more likely to be lexicalized. One would like to embed our neural networks inside of explicit models of language evolution to also corroborate this hypothesis. We hope to pursue all of these avenues in future research.

References

Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu & Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, 265–284. https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf.

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

Barwise, Jon & Robin Cooper. 1981. Generalized Quantifiers and Natural Language. *Linguistics and Philosophy* 4(2). 159–219. https://doi.org/10.1007/BF00350139.

Bengio, Yoshua. 1991. *Artificial Neural Networks and their Application to Sequence Recognition*. Montreal: McGill University dissertation. http://digitool.library.mcgill.ca/R/-?func=dbin-jump-full&object_id=70220.

van Benthem, Johan. 1984. Questions About Quantifiers. *The Journal of Symbolic Logic* 49(2). 443–466. https://doi.org/10.2307/2274176.

van Benthem, Johan. 1986. *Essays in Logical Semantics*. Dordrecht: D. Reidel Publishing Company. https://doi.org/10.1007/978-94-009-4540-1.

60 Another natural candidate is EXTENSIONALITY, which says that the carrier set of the model is in some ways irrelevant: if $A, B \subseteq M \subseteq M'$, then $\langle M, A, B\rangle \in \mathsf{Q}$ if and only if $\langle M', A, B\rangle \in \mathsf{Q}$. We find it plausible that this universal will pattern with CONSERVATIVITY in not being explainable by our model, but note that the two go hand-in-hand: a type $\langle 1,1\rangle$ quantifier satisfies both universals exactly when it is the *relativization* of a type $\langle 1\rangle$ quantifier (Peters & Westerståhl 2006, pp. 141-143). This fully captures the asymmetric role played by the restrictor of a determiner.

61 See Gärdenfors 2014 for more on convexity, and Chemla et al. 2018 for more on the connection to monotonicity.

Bolukbasi, Tolga, Kai-Wei Chang, James Zou, Venkatesh Saligrama & Adam Kalai. 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In *Advances in Neural Information Processing Systems (NIPS 29)*, https://papers.nips.cc/paper/6228-man-is-to-computer-programmer-as-woman-is-to-homemaker-debiasing-word-embeddings.pdf.

Buolamwini, Joy & Timnit Gebru. 2018. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Conference on Fairness, Accountability and Transparency, in Proceedings of Machine Learning Research*, vol. 81, 77–91. http://proceedings.mlr.press/v81/buolamwini18a/buolamwini18a.pdf.

Chemla, Emmanuel, Brian Buccola & Isabelle Dautriche. 2018. Connecting content and logical words. Accepted at *Journal of Semantics*. https://semanticsarchive.net/Archive/WVhYzUwM/Chemla-Buccola-Dautriche-ConnectWords.pdf.

Chomsky, Noam. 1965. *Aspects of the Theory of Syntax*. Cambridge: The MIT Press.

Chomsky, Noam. 1980. *Rules and Representations*. Oxford: Basil Blackwell.

Croft, William. 1990. *Typology and Universals*. Cambridge: Cambridge University Press.

Dixon, R.M.W. 1977. Where have all the adjectives gone? *Studies in Language* 1(1). 19–80. https://doi.org/10.1075/sl.1.1.04dix.

Donahue, Jeff, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko & Trevor Darrell. 2017. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(4). 677–691. https://doi.org/10.1109/TPAMI.2016.2599174.

Elman, Jeffrey L. 1990. Finding Structure in Time. *Cognitive Science* 14(2). 179–211. https://doi.org/10.1207/s15516709cog1402_1.

Feldman, Jacob. 2000. Minimization of Boolean Complexity in Human Concept Learning. *Nature* 407(05 October 2000). 630–633. https://doi.org/10.1038/35036586.

von Fintel, Kai. 1997. Bare Plurals, Bare Conditionals, and *Only*. *Journal of Semantics* 14(1). 1–56. https://doi.org/10.1093/jos/14.1.1.

von Fintel, Kai & Edward L Keenan. 2018. Determiners, Conservativity, Witnesses. *Journal of Semantics* 35(1). 207–217. https://doi.org/10.1093/jos/ffx018.

von Fintel, Kai & Lisa Matthewson. 2008. Universals in semantics. *The Linguistic Review* 25. 139–201. https://doi.org/10.1515/TLIR.2008.004.

Fox, Danny. 2002. Antecedent-Contained Deletion and the Copy Theory of Movement. *Linguistic Inquiry* 33(1). 63–96. https://doi.org/10.1162/002438902317382189.

Gärdenfors, Peter. 2014. *The Geometry of Meaning*. The MIT Press.

Gierasimczuk, Nina. 2007. The Problem of Lerning the Semantics of Quantifiers. In Balder ten Cate & Henk Zeevat (eds.), *6th International Tbilsi Symposium on Logic, Language, and Computation (TbiLLC)*, vol. 4363 Lecture Notes in Computer Science, 117–126. https://doi.org/10.1007/978-3-540-75144-1_9.

Gierasimczuk, Nina. 2009. Identification through Inductive Verification: Application to Monotone Quantifiers. In P Bosch, D Gabelaia & J Lang (eds.), *7th International Tbilsi Symposium on Logic, Language, and Computation (TbiLLC)*, vol. 5422 Lecture Notes in Artificial Intelligence, 193–205. https://doi.org/10.1007/978-3-642-00665-4_16.

Gisiger, Thomas & Mounir Boukadoum. 2011. Mechanisms gating the flow of information in the cortex: what they might look like and what their uses may be. *Frontiers in Computational Neuroscience* 5(1). 1–15. https://doi.org/10.3389/fncom.2011.00001.

Goodfellow, Ian, Yoshua Bengio & Aaron Courville. 2016. *Deep Learning*. The MIT Press. https://www.deeplearningbook.org/.

Goodman, Nelson. 1955. *Fact, Fiction, & Forecast*. Cambridge, MA: Harvard University Press.

Graves, Alex. 2013. Generating Sequences With Recurrent Neural Networks. https://arxiv.org/abs/1308.0850.

Hackl, Martin. 2009. On the grammar and processing of proportional quantifiers: *most* versus *more than half*. *Natural Language Semantics* 17(1). 63–98. https://doi.org/10.1007/s11050-008-9039-x.

He, Haibo & Edwardo A. Garcia. 2009. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering* 21(9). 1263–1284. https://doi.org/10.1109/TKDE.2008.239.

Hengeveld, Kees, Jan Rijkhoff & Anna Siewierska. 2004. Parts-of-speech systems and word order. *Journal of Linguistics* 40(3). 527–570. https://doi.org/10.1017/S0022226704002762.

Hochreiter, Sepp & Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9(8). 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Hunter, Tim & Jeffrey Lidz. 2013. Conservativity and learnability of determiners. *Journal of Semantics* 30(3). 315–334. https://doi.org/10.1093/jos/ffs014.

Hyman, Larry M. 2008. Universals in phonology. *The Linguistic Review* 25(1-2). 83–137. https://doi.org/10.1515/TLIR.2008.003.

Jordan, Michael I. 1986. Serial Order: a Parallel Distributed Processing Approach. Tech. Rep. ICS-8604. https://pdfs.semanticscholar.org/f8d7/7bb8da085ec419866e0f87e4efc2577b6141.pdf.

Jozefowicz, Rafal, Oriol Vinyals, Mike Schuster, Noam Shazeer & Yonghui Wu. 2016. Exploring the Limits of Language Modeling. https://arxiv.org/abs/1602.02410.

Katsos, Napoleon, Chris Cummins, Maria-José Ezeizabarrena, Anna Gavarró, Jelena Kuvač Kraljević, Gordana Hrzica, Kleanthes K. Grohmann, Athina Skordi, Kristine Jensen de López, Lone Sundahl, Angeliek van Hout, Bart Hollebrandse, Jessica Overweg, Myrthe Faber, Margreet van Koert, Nafsika Smith, Maigi Vija, Sirli Zupping, Sari Kunnari, Tiffany Morisseau, Manana Rusieshvili, Kazuko Yatsushiro, Anja Fengler, Spyridoula Varlokosta, Katerina Konstantzou, Shira Farby, Maria Teresa Guasti, Mirta Vernice, Reiko Okabe, Miwa Isobe, Peter Crosthwaite, Yoonjee Hong, Ingrida Balčiunienė, Yanti Marina Ahmad Nizar, Helen Grech, Daniela Gatt, Win Nee Cheong, Arve Asbjørnsen, Janne von Koss Torkildsen, Ewa Haman, Aneta Mi ekisz, Natalia Gagarina, Julia Puzanova, Darinka Anđelković, Maja Savić, Smiljana Jošić, Daniela Slančová, Svetlana Kapalková, Tania Barberán, Duygu Özge, Saima Hassan, Cecilia Yuet Hung Chan, Tomoya Okubo, Heather van der Lely, Uli Sauerland & Ira Noveck. 2016. Cross-linguistic patterns in the acquisition of quantifiers. *Proceedings of the National Academy of Sciences* 113(33). 9244–9249. https://doi.org/10.1073/pnas.1601341113.

Keenan, Edward L. & Denis Paperno (eds.). 2012. *Handbook of Quantifiers in Natural Language*, vol. 90 Studies in Linguistics and Philosophy. Dordrecht: Springer Netherlands. https://doi.org/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). 253–326. https://doi.org/10.1007/BF00630273.

Keenan, Edward L & Dag Westerståhl. 1997. Generalized Quantifiers in Linguistics and Logic. In Johan van Benthem & Alice ter Meulen (eds.), *Handbook of Logic and Language*, 837–893. Elsevier. https://doi.org/10.1016/B978-044481714-3/50020-5.

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

Kingma, Diederik P. & Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference of Learning Representations (ICLR)*, https://arxiv.org/abs/1412.6980.

Kotek, Hadas, Edwin Howard, Yasutada Sudo & Martin Hackl. 2011a. Three Readings of *Most*. In Neil Ashton, Anca Chereches & David Lutz (eds.), *Semantics and Linguistic Theory (SALT 21)*, 353–372. https://doi.org/10.3765/salt.v0i0.2621.

Kotek, Hadas, Yasutada Sudo, Edwin Howard & Hackl. 2011b. *Most* Meanings are Superlative. In *Experiments at the Interfaces*, vol. 37 (Syntax and Semantics 2011), 101–145. Emerald Group Publishing. https://doi.org/10.1108/S0092-4563(2011)0000037008.

Lindström, Per. 1966. First order predicate logic with generalized quantifiers. *Theoria* 32(3). 186–195. https://doi.org/10.1111/j.1755-2567.1966.tb00600.x.

May, Robert. 1991. Syntax, Semantics, and Logical Form. In Asa Kasher (ed.), *The Chomskyan Turn*, Blackwell Publishers.

McMillan, Corey T, Robin Clark, Peachie Moore, Christian Devita & Murray Grossman. 2005. Neural basis for generalized quantifier comprehension. *Neuropsychologia* 43(12). 1729–1737. https://doi.org/10.1016/j.neuropsychologia.2005.02.012.

McNab, Fiona & Torkel Klingberg. 2008. Prefrontal cortex and basal ganglia control access to working memory. *Nature Neuroscience* 11(1). 103–107. https://doi.org/10.1038/nn2024.

Mostowski, Andrzej. 1957. On a generalization of quantifiers. *Fundamenta Mathematicae* 44(2). 12–36.

Newmeyer, Frederick J. 2008. Universals in syntax. *The Linguistic Review* 25(1-2). 35–82. https://doi.org/10.1515/TLIR.2008.002.

Nielsen, Michael A. 2015. *Neural Networks and Deep Learning*. Determination Press. http://neuralnetworksanddeeplearning.com/.

Paperno, Denis & Edward L. Keenan (eds.). 2017. *Handbook of Quantifiers in Natural Language: Volume II*, vol. 97 Studies in Linguistics and Philosophy. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-44330-0.

Partee, Barbara H. 1992. Syntactic Categories and Semantic Type. In Michael Rosner & Roderick Johnson (eds.), *Computational Linguistics and Formal Semantics*, 97–126. Cambridge University Press.

Peters, Stanley & Dag Westerståhl. 2006. *Quantifiers in Language and Logic*. Oxford: Clarendon Press.

Peters, Stanley & Dag Westerståhl. 2013. The semantics of possessives. *Language* 89(4). 713–759. https://doi.org/10.1353/lan.2013.0065.

Pezzelle, Sandro, Marco Marelli & Raffaella Bernardi. 2017. Be Precise or Fuzzy: Learning the Meaning of Cardinals and Quantifiers from Vision. In *European Chapter of the Association for Computational Linguistics (EACL 15)*, 337–342. http://arxiv.org/abs/1702.05270.

Piantadosi, Steven T. 2013. Modeling the acquisition of quantifier semantics: a case study in function word learnability http://colala.bcs.rochester.edu/papers/piantadosi2012modeling.pdf.

Piantadosi, Steven T, Joshua B Tenenbaum & Noah D Goodman. 2016. The logical primitives of thought: Empirical foundations for compositional cognitive models. *Psychological Review* 123(4). 392–424. https://doi.org/10.1037/a0039980.

Pullum, Geoffrey K. & Barbara C. Scholz. 2002. Empirical assessment of stimulus poverty arguments. *The Linguistic Review* 18(1-2). 9–50. https://doi.org/10.

1515/tlir.19.1-2.9.

Romero, Maribel. 2015. The conservativity of *many*. In Thomas Brochhagen, Floris Roelofsen & Nadine Theiler (eds.), *20th amsterdam colloquium*, 20–29. http://ling.uni-konstanz.de/pages/home/romero/papers/ho-romero_v6-AC20.pdf.

Romoli, Jacopo. 2015. A Structural Account of Conservativity. *Semantics-Syntax Interface* 2(1). 28–57. https://www.academia.edu/8736879/A_structural_account_of_conservativity_final_version_.

Rumelhart, David E, James L McClelland & The PDP Reserach Group. 1986a. *Parallel Distributed Processing*, vol. 1. The MIT Press.

Rumelhart, David E, James L McClelland & The PDP Reserach Group. 1986b. *Parallel Distributed Processing*, vol. 2. The MIT Press.

Solt, Stephanie. 2016. On measurement and quantification: The case of *most* and *more than half*. *Language* 92(1). 65–100. https://doi.org/10.1353/lan.2016.0016.

Sorodoc, Ionut, Angeliki Lazaridou, Gemma Boleda, Sandro Pezzelle & Raffaella Bernardi. 2016. "Look , some green circles!": Learning to quantify from images. In *5th Workshop on Vision and Language*, 75–79.

Sorodoc, Ionut, Sandro Pezzelle, Aurélie Herbelot, Mariella Dimiccoli & Raffaella Bernardi. 2018. Learning quantification from images: A structured neural architecture. *Natural Language Engineering* 24(3). 363–392. https://doi.org/10.1017/S1351324918000128.

Sportiche, Dominique. 2005. Division of labor between Merge and Move: Strict locality of selection and apparent reconstruction paradoxes. http://ling.auf.net/lingbuzz/000163.

Steinert-Threlkeld, Shane. 2016. Some Properties of Iterated Languages. *Journal of Logic, Language and Information* 25(2). 191–213. https://doi.org/10.1007/s10849-016-9239-6.

Steinert-Threlkeld, Shane. 2018. An Explanation of the Veridical Uniformity Universal. https://semanticsarchive.net/Archive/DI5ZTNmN/UniversalResponsiveVerbs.pdf.

Steinert-Threlkeld, Shane & Thomas F. Icard III. 2013. Iterating semantic automata. *Linguistics and Philosophy* 36(2). 151–173. https://doi.org/10.1007/s10988-013-9132-6.

Sutskever, Ilya, Oriol Vinyals & Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D Lawrence & Killian Q Weinberger (eds.), *Advances in Neural Information Processing Systems (NIPS 27)*, https://arxiv.org/abs/1409.3215.

Szabolcsi, Anna. 2010. *Quantification* Research Surveys in Linguistics. Cambridge: Cambridge University Press.

Szymanik, Jakub. 2016. *Quantifiers and Cognition: Logical and Computational Perspectives*, vol. 96 Studies in Linguistics and Philosophy. Springer. https:

//doi.org/10.1007/978-3-319-28749-2.

Szymanik, Jakub & Marcin Zajenkowski. 2010. Comprehension of simple quantifiers: empirical evaluation of a computational model. *Cognitive Science* 34(3). 521–532. https://doi.org/10.1111/j.1551-6709.2009.01078.x.

Tenenbaum, Joshua B, Charles Kemp, Thomas L Griffiths & Noah D Goodman. 2011. How to grow a mind: statistics, structure, and abstraction. *Science* 331(6022). 1279–1285. https://doi.org/10.1126/science.1192788.

Tiede, Hans-Joerg. 1999. Identifiability in the Limit of Context-Free Generalized Quantifiers. *Journal of Language and Computation* 1(1). 93–102.

Werbos, Paul J. 1988. Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks* 1(4). 339–356. https://doi.org/10.1016/0893-6080(88)90007-X.

Westerståhl, Dag. 1985. Logical constants in quantifier languages. *Linguistics and Philosophy* 8(4). 387–413. https://doi.org/10.1007/BF00637410.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes & Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. https://arxiv.org/abs/1609.08144.

Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel & Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Francis Bach & David Blei (eds.), *International Conference on Machine Learning (ICML 32)*, 2048–2057. https://arxiv.org/abs/1502.03044.

Shane Steinert-Threlkeld
Science Park 107
1098 XG Amsterdam, Netherlands
S.N.M.Steinert-Threlkeld@uva.nl

Jakub Szymanik
Science Park 107
1098 XG Amsterdam, Netherlands
J.K.Szymanik@uva.nl