No Free Lunch in Linguistics or Machine Learning

Jonathan Rawski & Jeffrey Heinz*
Dept. of Linguistics
Institute for Advanced Computational Science
Stony Brook University

To Appear in *Language*

Abstract

Pater's target article proposes that neural networks will provide theories of learning that generative grammar lacks. We argue that his enthusiasm is premature since the biases of neural networks are largely unknown, and he disregards decades of work on machine learning and learnability. Learning biases form a two-way street: all learners have biases, and those biases constrain the space of learnable grammars in mathematically measurable ways. Analytical methods from the related fields of computational learning theory and grammatical inference allow one to study language learning, neural networks, and linguistics at an appropriate level of abstraction. The only way to satisfy our hunger and to make progress on the science of language learning is to confront these core issues directly.

1 Introduction

Pater (2018:1) claims that "generative linguistics is unlikely to fulfill its promise of accounting for language learning if it continues to maintain its distance from neural and statistical approaches to learning." It is a crucial time to address this issue, given the recent splash of machine learning success in various engineering domains and the renewed interest in generative structure in natural language processing (NLP). At the same time, there is a sense that the Machine Learning Revolution has morphed more into engineering than science (one recent machine learning plenary talk called it "alchemy" and longed for "the rigor police" to return (Rahimi & Recht 2017)). Far from discouraging us, this presents a unique challenge for linguistics. How can linguistics inform the theory of what is learnable within any system, and how can insights from learning aid the theoretical goals of linguistics, and by extension, cognitive science?

Pater is right that interactions between these two fields presents a unique opportunity, but his enthusiasm for neural networks is premature. The real opportunity is to, as Mark

*We thank Alëna Aksënova, Aniello de Santo, Hossep Dolatian, and William Idsardi for helpful discussions.

Steedman (2008:144) says, "pull ourselves together as a discipline, lift our eyes above our rivalries, and take a longer view of where we are going (and where we have come from) than seems to be the current style." In this case, that means engaging the real and hard problems which underlie making linguistic generalizations from data. This involves understanding the biases of learning systems, which is not only about how rules and constraints are determined in the face of recalcitrant data, but also understanding how such biases can be ascertained and studied. Pater not only conflates ignorance of bias with absence of bias, but his narrative that neural networks solve the problem of "how systems that adequately represent linguistic knowledge can be learned" (p.26) disregards an extensive amount of research that addresses these hard problems directly.

2 Bias is a Two-way Street.

The main idea of this response is stated in the "Key Ideas in Machine Learning" chapter of the forthcoming second edition of (Mitchell 2017:4) (see also Wolpert & Macready 1997):

> When we consider it carefully, it is clear that no system - computer program or human - has any basis to reliably classify new examples that go beyond those it has already seen during training, unless that system has some additional prior knowledge or assumptions that go beyond the training examples. In short, there is no free lunch—no way to generalize beyond the specific training examples, unless the learner commits to some additional assumptions.

Furthermore, even in the presence of strong biases, the data is typically recalcitrant. Charles Yang (2013) notes, for instance, that for n-gram models, which are a strongly biased sequence model embedded in many modern NLP systems, the space of word combinations grows exponentially with n-gram size (m words yields m^n n-grams, so n^2 bigrams, n^3 trigrams, etc.). Despite the enormity of this space, if these combinations are uniformly distributed then learning is a breeze, since all admissible word combinations will sooner or later show up. However, Yang (2013:2) points out that "In the actual world, we are undercut by Zipf's long tail, which grows longer as the space of word combination gets larger" so that "in a million-word collection of English, 45% of words, 80% of word pairs, and a full 95% of word triples appear exactly once." Zipf's Law shows that most information about the language system as a whole resides in the long tail. For a learner, this means data is drastically underspecified: most things are rare, what data exists is unhelpful, and more data exacerbates the problem, rather than easing it. Despite Pater's enthusiasm, neural networks are not immune from these statistical realities. When examining natural language corpora, you may never get to see events that are rare but matter for accurately characterizing the system under observation.

Problems of this sort are known in linguistics as Poverty of the Stimulus, and in computer science as instances of the Data Sparsity Problem, and they are frequently used to necessitate strong learning biases. It is uncontroversial that human language acquisition must have some type of bias or innate component, often referred to as Universal Grammar (Nowak *et al.* 2001; Nowak *et al.* 2002). This is true even for data-driven statistical learning, since children often learn language in ways that defy adult distributions. In fact, without some learning priors, even the input distribution cannot be recovered, simply because distributions are based on the structure of their parameters (Lappin & Shieber 2007). The non-trivial open question central to all modern linguistic research instead concerns the NATURE of this "additional prior knowledge" or bias—where it lies on the scale from weak to strong, and how complex, uniform, and task-specific it is.

Despite these well-known challenges, Pater repeatedly dismisses Chomsky's arguments "in the absence of a specified theory of learning." He says that "the argument is not completely solid that a rich Universal Grammar (UG)... is necessary to explain language acquisition" (p.3) and later "it is much less clear that a successful theory will need pre-specified UG parameters or constraints." (p.3) Chomsky in fact addressed these sorts of comments in a famous debate on language and learning (Piattelli-Palmarini 1980:100), so we will let him speak for himself:

> Papert objects, quite incorrectly, that somehow the burden of proof is on me to go beyond saying that I see no way of doing something. Of course, that's not the case: if someone wants to show that some general developmental mechanisms exist (I don't know of any), then he has to do exactly what Rosenblatt (who invented the perceptron) did, namely, he has to propose the mechanism; if you think that sensorimotor mechanisms are going to play a role, if you can give a precise characterization of them I will be delighted to study their mathematical properties and see whether the scope of those mechanisms includes the case in question.

Pater clearly views neural networks as a counterpoint to "pre-specified UG parameters or constraints". He states "with the development of the rich theories of learning represented by modern neural networks, the learnability argument for a rich UG is particularly threatened" (p.3), and there are many statements like "From a generative perspective, an RNN is a very impoverished theory of UG. Not only does it lack parameters or constraints encoding the substance of possible grammatical systems, the basic structures...are themselves absent, at least prior to learning." (p.23) Then later he admits that "a [neural] network needs a specified structure for it to represent the effects of learning," (p.13) so he is trying to have it both ways, when he cannot.

In our view, Pater continually confuses ignorance of biases with absence of biases. The intrinsic biases of neural network models are almost completely unknown, and are not necessarily weak or impoverished (Dyer 2018). Additionally, the utility of a weak-bias learning method for NLP tasks does not guarantee explanatory adequacy.

So, how does the learner pay for lunch? The other, more important side of the two-way street is that the nature of particular biases and learning strategies have measurable effects on the targets of learning. Results of this sort, including Chomsky's, are claims about what is LEARNABLE, which is the domain of computational learning theory. Learning theory provides the science behind the engineering by addressing issues of learnability under various mathematical conditions: in particular, what successful learning means in different scenarios and what resources it requires in terms of computation and number of training examples. Results from computational learning theory all echo the "No Free Lunch" principle and the core of any generative theory of linguistic competence: learning requires a structured hypothesis space and the structural properties MATTER.

Computational learning theory is often grounded in the theory of formal languages. The mathematical nature of the subject is a natural and advantageous perspective, for two reasons. First, there is the link between formal languages and computation, which is deeply relevant to neural network research, going back to (McCulloch & Pitts 1943), a landmark work in artificial neural networks. Kleene (1956) developed regular expressions in an effort to characterize the abilities of these neural nets. As Perrin (1990:3) put it, McCulloch and Pitts presented "a logical model for the behaviour of nervous systems that turned out to be the model of a finite-state machine." Minsky (1967:55) later made the often-quoted statement "every finite-state machine is equivalent to, and can be simulated by, some neural net." This important link is missing in Pater's narrative.

Second, formal languages and grammars are arguably the core of all generative theories of language. Chomsky has been very clear on this point. In the same debate, (Piattelli-Palmarini 1980:101), he states,

> that is exactly what generative grammar has been concerned with for twenty-five years: the whole complicated array of structures beginning, let's say, with finite-state automata, various types of context-free or context-sensitive grammars, and various subdivisions of these theories of transformational grammars—these are all theories of proliferating systems of structures designed for the problem of trying to locate this particular structure, language, within that system. So there can't be any controversy about the legitimacy of that attempt.

A wholesale adoption of neural networks today is premature because it still leaves us without a "precise characterization" of language acquisition and learnability. In our view, the insights of computational learning theory provide a concrete way to study the two-way street of learning biases. Whereas Pater ignores No Free Lunch, we take it as the science of learning's natural starting point.

3 The View from Learning Theory.

3.1 Computational Theories of Language.

Let's quickly look at the menu du jour. The computational nature of language patterns and the grammars which generate them has been mathematically well-studied. The Chomsky Hierarchy (Chomsky 1959) divides all logically possible patterns into nested regions of complexity (see Fig. 1). These regions distinguish abstract, structural properties of grammars. They have multiple, convergent definitions, which distill the necessary properties of any machine, grammar, or device capable of recognizing or generating the strings comprising the pattern (Harrison 1978; Hopcroft *et al.* 1979). For example, to recognize the formal languages in the *regular* region, the memory resources required do not grow with the size of the input past some constant bound. While the Chomsky hierarchy was originally formulated for strings, modern developments extend such abstractions to characterize 'languages' of other objects, such as trees and graphs.

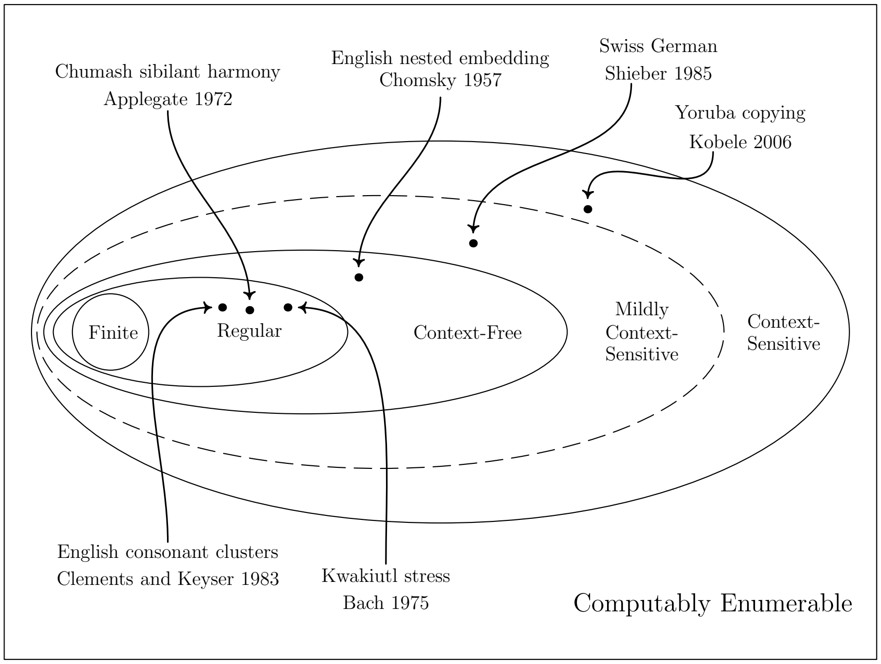

Figure 1: The Chomsky Hierarchy of string languages

From a more abstract perspective, every grammar is associated with a function, so the issue turns to the nature of that function. Such a function may map strings to either 0 or 1. We may also consider other functions which take strings as input and return various values, depending on the properties we want the grammar to have. For example, we may choose to make a grammar stochastic by replacing the binary mapping with a mapping to real values between 0 and 1, or to represent linguistic transformations by changing one representation into another. Some example functions are listed in Table 1 below, along with the linguistic intuition they encode.

Function	Description	Linguistic Correlate
$f : \Sigma^* \to \{0, 1\}$	Binary classification	(well-formedness)
$f : \Sigma^* \to [0, 1]$	Maps strings to real values	(gradient well-formedness)
$f : \Sigma^* \to \Delta^*$	Maps strings to strings	(single-valued transformation)
$f : \Sigma^* \to \wp(\Delta^*)$	Maps strings to sets of strings	(multi-valued transformation)

Table 1: Grammars as functions

Pater argues for a move toward statistical generalization in grammars and learning. There are stochastic formulations of each region in the Chomsky Hierarchy (Kornai 2011), but the critical properties of languages are STRUCTURAL. For example, "context-freeness" is independent of whether that function's co-domain is real or Boolean. This means that the addition of probabilities to a grammar formalism doesn't consequently increase its expressivity in this way. In other words, no amount of statistics can make a regular grammar context-free. (For more on the Chomsky Hierarchy and its implications to grammar learning and learning experiments, see Jäger & Rogers (2012).)

3.2 Learners as Functions to Grammars.

The basic question of whether some form in a language is well-formed according to a given grammar is called the membership problem. The membership problem allows us to clearly formulate a definition of learning, shown in Figure 2. In particular, is there a learning algorithm A which takes as input a finite subset D of the acceptable forms of a language S (a highly skewed sample, if Zipf's law holds) and returns a grammar which solves the membership problem M for that language? It is crucial that learners return a grammar, not a language. This is simply because, mathematically, grammars are of finite size, while the functional characterizations of patterns may be infinite in size.

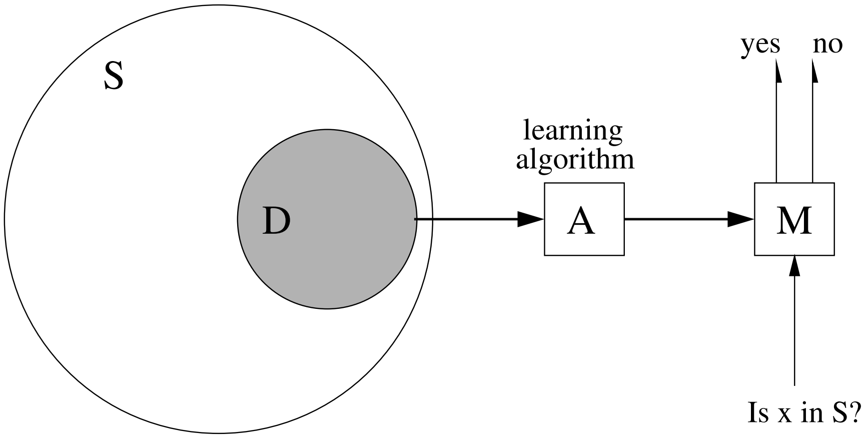

Figure 2: Learning to solve the membership problem from positive data.

Thus learners are best described as functions from experience to grammars. This characterization of learners is very precise, but broad, since any learning procedure can be thought of as a function from experience to grammars, including connectionist (Rumelhart & McClelland 1986), Bayesian (Griffiths *et al.* 2008), entropy-based learners (Goldwater & Johnson 2003), and learners situated in generative models (Wexler & Culicover 1980; Berwick 1985; Tesar & Smolensky 2000; Niyogi 2006).

Learning theory is also concerned with "the circumstances under which these hypotheses stabilize to an accurate representation of the environment from which the evidence is drawn. Such stability and accuracy are conceived as the hallmarks of learning" (Osherson *et al.* 1986). Gold (1967) introduced one definition for "successful learning" and showed that no major classes in the Chomsky hierarchy except the finite languages are learnable "in the limit" from adversarial presentations[1] of positive examples, but the computably enumerable (and by extension context-free and regular) are learnable from adversarial presentations of positive and negative examples. The Probably Approximately Correct (PAC) framework (Valiant 1984; Kearns *et al.* 1987; Angluin 1988) differs from Gold learning in that the required training data and computation must be feasible (i.e. polynomial), and is also less strict, since it loosens the definition of success from exact learning to likely approximate learning. However, not even the finite languages are PAC-learnable. The important thing to keep in mind is that mathematical definitions clarify the difficulty of feasibly learning

[1] This is a technical term. See Heinz (2016) and Clark & Lappin (2011) for details.

classes of formal languages that resemble natural languages in key respects. However, many structured classes which cross-cut the Chomsky hierarchy and exclude some finite languages are feasibly learnable in various senses, as we will discuss below.

In general, the process of inferring the rules of the grammar from samples of data is called Grammatical Inference (de la Higuera 2010; Heinz *et al.* 2015; Heinz & Sempere 2016), the branch of machine learning arguably most relevant for generative linguistics. The clarity given by the methods developed within the Grammatical Inference tradition is that the biases are analytically transparent, so beyond beyond eating lunch, we know who paid for it. There is no debt. Such analysis carries important implications for typological coverage, which we now discuss.

3.3 Typological Consequences of Learning Theories.

What kind of a lunch can we now afford? Theories of language learning make empirical typological predictions, which goes to the issue of whether languages vary systematically or not. We divide these predictions into three types, according to the learning strategy. The first type of learner assumes no structural variations. A second assumes one unified learning mechanism. The third has a modular approach which ties the learning strategy to the grammars being learned. Here we showcase the limitations of each and implications for learning with neural nets.

The first strategy is simply to go for broke and learn everything. This is the strategy pursued by Chater & Vitányi (2007), who use Minimum Description Length principles to prove that in a particular learning setting, any computably enumerable distribution can be learned from positive evidence. A caveat to this strategy is that while it is possible in principle, it is very much infeasible in practice. Additionally, it predicts that any pattern is learnable with enough data, which seems linguistically incorrect since generalizations attested in the world's languages are not arbitrary and much more restricted than being computable. Many people using neural networks for instance believe that they are domain-general in exactly this way, and no pattern is beyond their reach.

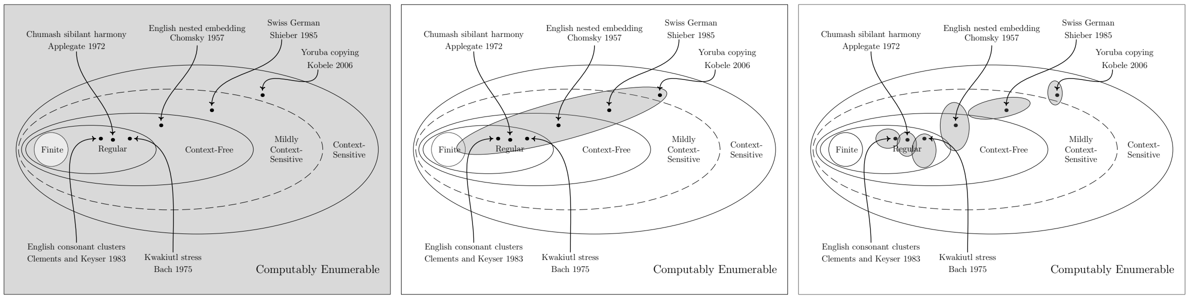

Figure 3: A universal pattern learner (left) vs. a structured, domain-general learner (center) vs. multiple modular learners (right).

The second strategy is similarly "domain-general" but constrained: a single learner which cross-cuts the Chomsky hierarchy but does not exhaust any one class (Figure 3, left). Such a strategy is pursued by Clark (2010), among others. The caveat with this, noted by (Heinz & Idsardi 2011; Heinz & Idsardi 2013), is that it predicts that the sorts of dependencies seen in syntax at the sentence-level should also be seen at the word level, in phonology and morphology, while the reality is distinctly the opposite. Supra-regular patterns abound in syntactic dependencies, while they are conspicuously absent in morphophonology (Figure 4).

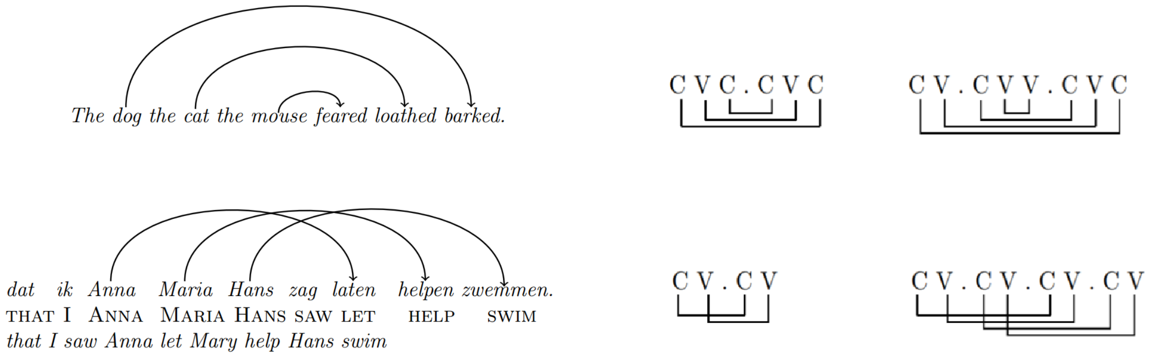
The dog the cat the mouse feared loathed barked.
C V C . C V C
C V . C V V . C V C
dat ik Anna Maria Hans zag laten helpen zwemmen.
THAT I ANNA MARIA HANS SAW LET HELP SWIM
that I saw Anna let Mary help Hans swim
C V . C V
C V . C V . C V . C V

Figure 4: Left: Nested embedding in English(top), and crossing dependencies in Dutch (bottom), both supra-regular syntactic dependencies. Right: Nested embedding (top) and crossing dependencies (bottom) in phonology, both unattested (adapted from Heinz & Idsardi 2013).

The final strategy is a "modular" learner, with distinct hypothesis spaces for each of the distinct subclasses of grammars that characterize the patterns found in natural language (Figure 3, right). Learning algorithms in phonology are successful because the generalization strategies employed do not consider every finite pattern nor do they extend beyond the regular boundary, while syntactic learners are successful when the learners' generalizations are constrained to the right, non-superfinite classes of nonregular patterns.

Where do neural networks and their learning algorithms fit in this picture? The crucial error that is regularly made in applying neural networks to structured problems of this kind is to confuse ignorance of biases with absence of biases. All neural network architectures face the two-way street of bias, simply because there is No Free Lunch. It is unfortunate that we have no idea what those biases are, so its practice is bound to resemble "alchemy." Ignoring this will be a detriment to the increased interaction Pater proposes since, as this section showed, learning biases have concrete typological consequences, which extend to theories of cognition. Any serious scientific application of neural architectures within linguistics must always strive to make the learner's biases transparent.

4 Grammatical Inference and Neural Networks.

By clearly knowing how to pay for lunch, we are bound to have many interesting and satisfying meals. The equivalence between language and automata classes, and the large variety of neural architectures used to imitate the behavior of sequential machines, allow us to refine grammatical inference questions to neural network generalization. Can a neural network of architecture class A perform the same computation as an automaton of class M? Can a network of A be a recognizer for languages in class C? Can a network of A be trained to perform the same computation as an automaton of M from a training set, or to recognize a language of C from a training set? Can a grammatical inference algorithm for language class C unpack the generalization mechanism of a neural architecture trained on C? Here we highlight two recent studies that explore these questions AT THE SIMPLEST LEVEL with striking results.

One example comes from Avcu *et al.* 2017. They compared a class of Long Short-Term Memory (LSTM) architectures' performance on two well-understood subclasses of the regular languages — namely, the Strictly Local and Strictly Piecewise languages, which have been argued to be characteristic of certain local and long-distance dependencies in natural language (Heinz 2010; Heinz 2018). They report that their LSTMs performed above chance in all experiments, but surprisingly the LSTMs had particular difficulty in the simplest Strictly Local language and the most complex Strictly Piecewise language. When learning fails it is natural to ask whether the data was sufficient. They used the grammatical inference learning algorithm Regular Positive and Negative Inference (RPNI, (Oncina & Garcia 1992)) to test this, since RPNI essentially solves the problem of efficiently learning regular languages from positive and negative data. Testing with RPNI, they determined that the data were in fact mostly sufficient in both Strictly Local and Strictly Piecewise cases because RPNI mostly returned the exact target grammar with the same data with which the LSTM failed to generalize correctly. The possibility that formal properties of different kinds of local and non-local dependencies present in the data affects the behavior of neural networks is exciting, and shows the fruitfulness of studying neural network and deep learning models in the light of formal language theory and grammatical inference algorithms.

Weiss *et al.* (2017) present another striking result combining methods from grammatical inference with neural network and deep learning architectures. They use Angluin (1987)'s L^* algorithm to test whether one can extract from a trained recurrent neural network (RNN) a deterministic finite-state automaton (DFA) that classifies sequences in a manner observably equivalent to the RNN. They show that in contrast to other extraction methods, they are able to find smaller, more accurate DFAs which model the RNN in a fraction of the time.

However, what is most interesting and relevant in this study are the generalization errors in the trained RNNs this method allows them to discover. Weiss *et al.* (2017:8) note such cases are "annoyingly frequent: for many RNN-acceptors with 100% train and test accuracy on large test sets, our method was able to find many simple misclassified examples." They state this reveals the "brittleness in generalization" of trained RNNs, and they suggest that evidence based on test-set performance "should be interpreted with extreme caution."

These two studies are by no means alone (see Ayache *et al.* 2018 for another example), but they drive home the point that has been made throughout this piece: no matter the domain of learning, and no matter how the architecture of the learner is structured, there is No Free Lunch. NNs represent a powerful engineering tool, but to understand we require computational analysis at the abstract level that formal language theory and learning theory provide. By incorporating these insights, the "black-box" nature of neural networks and deep learning methods that are the cause of so much of the skepticism surrounding their application can be opened up and explored in a principled way. This offers a way for the features of a particular learning domain, say that of language, to inform theories of neural networks as a whole, by working at the right level of abstraction like Kleene did over sixty years ago.

5 Conclusion.

This meal has been short, but hopefully now we are not perplexed when the waiter comes. Pater's article continually ignores the hard lesson from computational learning theory: there is No Free Lunch, and you pay for your lunch with theory. Learning biases form a two-way street: no learning system is bias-free, and the nature of the biases have measurable effects on the targets of learning. Neural networks are not immune from the two-way street, and confusing our ignorance of the nature of their biases with an absence of biases will only impede the progress made on linguistic learning in the last 60 years. However, taking these issues seriously will allow linguistics to sharpen its view of learning, and to contribute in a meaningful way to the larger science of computational and machine learning. Computational learning theory and grammatical inference have shown some plausible paths forward on this journey, and we are confident that by paying careful attention to hard problems, we will be rewarded with a smorgasbord of solutions.

References

Angluin, Dana. 1987. Learning regular sets from queries and counterexamples. *Information and Computation* 75.87–106.

——. 1988. Queries and concept learning. *Machine Learning* 2.319–342.

Applegate, Richard B., 1972. *Ineseño Chumash grammar*. University of California,Berkeley dissertation.

Avcu, Enes, Chihiro Shibata, & Jeffrey Heinz. 2017. Subregular complexity and deep learning. In *CLASP Papers in Computational Linguistics: Proceedings of the Conference on Logic and Machine Learning in Natural Language (LaML 2017), Gothenburg, 12 –13 June*, ed. by Simon Dobnik & Shalom Lappin, 20–33.

Ayache, Stéphane, Rémi Eyraud, & Noé Goudian. 2018. Explaining black boxes on sequential data using weighted automata. In *Proceedings of the 14th International Conference on Grammatical Inference*, ed. by Olgierd Unold, Witold Dyrka, & Wojciech Wieczorek, volume 93 of *Proceedings of Machine Learning Research*, 81–103. PMLR.

Bach, Emmon W. 1975. Long vowels and stress in Kwakiutl. In *Texas Linguistic Forum*, volume 2, 9–19.

Berwick, Robert. 1985. *The acquisition of syntactic knowledge*. Cambridge, MA: MIT Press.

Chater, Nick, & Paul Vitányi. 2007. 'Ideal learning' of natural language: Positive results about learning from positive evidence. *Journal of Mathematical Psychology* 51.135–163.

Chomsky, Noam. 1956. Three models for the description of language. *IRE Transactions on information theory* 2.113–124.

——. 1959. On certain formal properties of grammars. *Information and Control* 2.137–167.

Clark, Alexander. 2010. Three learnable models for the description of language. In *Language and Automata Theory and Applications, Fourth International Conference, LATA 2010*, ed. by Carlos Martín-Vide Adrian-Horia Dediu, Henning Fernau, LNCS, 16–31. Springer.

——, & Shalom Lappin. 2011. *Linguistic Nativism and the Poverty of the Stimulus*. Wiley-Blackwell.

Clements, George N., & Samuel Jay Keyser. 1983. *CV Phonology: A Generative Theory of the Syllable*. Cambridge, Mass.: MIT Press.

de la Higuera, Colin. 2010. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press.

Dyer, Chris, 2018. Recurrent neural networks and bias in learning natural languages. 1st Meeting of the Society for Computation in Linguistics.

Gold, E. Mark. 1967. Language identification in the limit. *Information and Control* 10.447–474.

Goldwater, Sharon, & Mark Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. In *Proceedings of the Stockholm Workshop on Variation within Optimality Theory*, ed. by Jennifer Spenader, Anders Eriksson, & Osten Dahl, 111–120.

Griffiths, Thomas, Charles Kemp, & Joshua Tenenbaum. 2008. Bayesian models of cognition. In *The Cambridge handbook of computational cognitive modeling*, ed. by Ron Sun. Cambridge University Press.

Harrison, Michael A. 1978. *Introduction to Formal Language Theory*. Addison-Wesley Publishing Company.

Heinz, Jeffrey. 2010. Learning long-distance phonotactics. *Linguistic Inquiry* 41.623–661.

——. 2016. Computational theories of learning and developmental psycholinguistics. In *The Oxford Handbook of Developmental Linguistics*, ed. by Jeffrey Lidz, William Synder, & Joe Pater, chapter 27, 633–663. Oxford, UK: Oxford University Press.

——. 2018. The computational nature of phonological generalizations. In *Phonological Typology*, ed. by Larry Hyman & Frans Plank, Phonetics and Phonology, chapter 5, 126–195. De Gruyter Mouton.

——, Colin de la Higuera, & Menno van Zaanen. 2015. *Grammatical Inference for Computational Linguistics*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

——, & William Idsardi. 2011. Sentence and word complexity. *Science* 333.295–297.

——, & William Idsardi. 2013. What complexity differences reveal about domains in language. *Topics in Cognitive Science* 5.111–131.

——, & José Sempere (eds.) 2016. *Topics in Grammatical Inference*. Springer-Verlag Berlin Heidelberg.

Hopcroft, John, Rajeev Motwani, & Jeffrey Ullman. 1979. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley.

Jäger, Gerhard, & James Rogers. 2012. Formal language theory: Refining the Chomsky hierarchy. *Philosophical Transactions of the Royal Society B* 367.1956–1970.

Kaplan, Ronald M., & Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20.331–378.

Kearns, Michael, Ming Li, Leonard Pitt, & Leslie Valiant. 1987. On the learnability of Boolean formulae. In *Proceedings of the nineteenth annual ACM Symposium on Theory of Computing, New York City, May 25–27, 1987*, ed. by A. V. Aho, 285–295, New York, NY 10036, USA. ACM Press.

Kleene, Stephen. 1956. Representation of events in nerve nets. In *Automata Studies*, ed. by C.E. Shannon & J. McCarthy, 3–40. Princeton University. Press.

Kobele, Gregory, 2006. *Generating Copies: An Investigation into Structural Identity in Language and Grammar*. University of California, Los Angeles dissertation.

Kornai, András. 2011. Probabilistic grammars and languages. *Journal of Logic, Language, and Information* 20.317–328.

Lappin, Shalom, & Stuart M Shieber. 2007. Machine learning theory and practice as a source of insight into universal grammar. *Journal of Linguistics* 43.393–427.

McCulloch, Warren S, & Walter Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* 5.115–133.

Minsky, Marvin L. 1967. *Computation: finite and infinite machines*. Prentice-Hall, Inc.

Mitchell, Tom. 2017. Key ideas in machine learning. In *Machine Learning: Second Edition*. (forthcoming) `http://www.cs.cmu.edu/~tom/mlbook/keyIdeas.pdf`.

Niyogi, Partha. 2006. *The Computational Nature of Language Learning and Evolution*. Cambridge, MA: MIT Press.

Nowak, Martin A., Natalia L. Komarova, & Partha Niyogi. 2001. Evolution of universal grammar. *Science* 291.114–118.

——, ——, & ——. 2002. Computational and evolutionary aspects of language. *Nature* 417.611–617.

Oncina, Jose, & Pedro Garcia. 1992. Identifying regular languages in polynomial time. In *Advances In Structural And Syntactic Pattern Recognition, Volume 5 Of Series In Machine Perception And Artificial Intelligence*, 99–108. World Scientific.

Osherson, Daniel, Scott Weinstein, & Michael Stob. 1986. *Systems that Learn*. Cambridge, MA: MIT Press.

Pater, Joe. 2018. Generative linguistics and neural networks at 60: Foundation, friction, and fusion. *Language* .

Perrin, Dominique. 1990. Finite automata. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics.*, ed. by J. van Leeuwen, 1–57. Elsevier.

Piattelli-Palmarini, Massimo (ed.) 1980. *Language and learning : the debate between Jean Piaget and Noam Chomsky*. Harvard University Press Cambridge, Mass.

Rahimi, Ali, & Ben Recht, 2017. Reflections on random kitchen sinks, `http://www.argmin.net/2017/12/05/kitchen-sinks/`.

Rumelhart, D. E., & J. L. McClelland. 1986. On learning the past tenses of English verbs. In *Parallel Distributed Processing, volume 2*, ed. by J.L. McClelland & D. E. Rumelhart, 216–271. Cambridge MA: MIT Press.

Shieber, Stuart. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy* 8.333–343.

Steedman, Mark. 2008. On becoming a discipline. *Computational Linguistics* 34.137–144.

Tesar, Bruce, & Paul Smolensky. 2000. *Learnability in Optimality Theory*. MIT Press.

Valiant, Leslie. 1984. A theory of the learnable. *Communications of the ACM* 27.1134–1142.

Weiss, Gail, Yoav Goldberg, & Eran Yahav. 2017. Extracting automata from recurrent neural networks using queries and counterexamples. *arXiv preprint arXiv:1711.09576* .

Wexler, Kenneth, & Peter Culicover. 1980. *Formal Principles of Language Acquisition*. MIT Press.

Wolpert, David H, & William G Macready. 1997. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation* 1.67–82.

Yang, Charles. 2013. Who's afraid of George Kingsley Zipf? or: Do children and chimps have language? *Significance* 10.29–34.