Bonus material for *The Laws of Thought*

Tom Griffiths

Thank you for reading *The Laws of Thought*. There were a few extra sections that didn't make it into the final version of the book. I'm providing them here as bonus material for dedicated readers.

Chapter 4

The Chomsky hierarchy

The Chomsky hierarchy also turned out to be important in computer science. So far, we have been focused on generating sentences, but there is a parallel problem of checking whether a sentence belongs to a particular language. For example, if you want to create a programming language for a computer, you need to be able to check that a particular piece of code is a valid expression in that language. It turns out that for languages in the Chomsky hierarchy, this can often be done using a device much simpler than a Turing machine.[1] To check whether a sentence is in a finite state language, we can just play along using the boardgame version of the grammar: begin in the start position, then as each word comes in check that there is a board position we can move to that is consistent with that word, and repeat until we either get an invalid word or reach the end of the game. This can be done by a machine that has one state for each board position and moves to the right along the sentence, considering each word in turn. This simple machine is known as a finite state automaton. Context-free languages can be checked by a machine that adds a kind of memory to this system, called a stack. It can push words on the top of the stack and pop them off again, but it can only look at the word on the top of the stack. This is called a pushdown automaton.Finally, context-sensitive languages can be checked by a Turing machine that has a tape of limited size, called a linear bounded Turing machine. Just as each class of languages is a subset of the next, each kind of machine is strictly less powerful than the next.

So, here's the full Chomsky hierarchy, including languages and machines:

Finite-state languages	Finite state automaton
Context-free languages	Pushdown automaton
Context-sensitive languages	Linear bounded Turing machine

This has been an intense chapter – grammars, automata, how can you keep track? Fortunately, when we were in graduate school my wife Tania Lombrozo and I wrote a short song about the Chomsky hierarchy that covers all you need to know. It's called "If language were finite" and is intended to be sung to the tune of "If I were a rich man" from *Fiddler on the Roof*:

If language were finite,
One could memorize
All sentences as if they were just lists.
But it's not. True novelty exists.
Language is no finite sys-tem.

A finite state grammar
Is a tempting second thought
But clearly isn't what we've got
The first words in almost any phrase
Can constrain the end in many ways

So how about a push-down automaton
Might that be the very thing?
Just like a finite-state with a proper stack.
There would be just one symbol popped from the top
Plus one as input from the string
And others there just waiting to pop back.

But hu-man language cannot be context-free
Swiss-German shows why this is true
Thanks to the cross seri-al de-pen-den-cy
So much for the very thought that language could be finite
I've shown why the notion just won't do
But now onto what language has to be...
Oy!

Language isn't finite
Nor is it finite state
Or even possibly push-down
Nor just strings composed of verb and noun
Language is a complex sys-tem.
There are transformations
Over parse-trees that you
simply cannot code as linear strings
Every sentence is hierarchical
With deep structure that can be revealed.

Plato's problem

Plato was one of the great philosophers of Ancient Greece. He studied with Socrates, and founded the Academy in Athens where many other philosophers studied with him in turn. One of those students was Aristotle, who put us on the path to understanding logic in Chapter 2.

Plato presented his philosophical ideas in the form of dialogues. These dialogues were conversations between characters who could present different perspectives on an idea. Sometimes,

Plato would use his teacher Socrates as one of the characters. The conversational format meant that one character could be brought to a realization by the other asking a series of questions – an approach that Plato had learned from his teacher, which we now refer to as the Socratic method.

In one famous dialogue Socrates has a conversation with a young aristocrat, Meno, about the origins of knowledge. The two are discussing the nature of virtue when Socrates happens to mention that what most people call learning is really recollection. Meno, understandably, picks up on this and asks what Socrates means by it, so Socrates stages a dramatic demonstration. He beckons over one of Meno's servants – a young boy – and starts drawing geometric figures and asking questions about them. With some astute questioning the servant comes to understand the relationship between the length of the side of a square and its area, and even rediscovers a version of the Pythagorean theorem – a fairly sophisticated piece of the mathematics curriculum in Ancient Greece.

Socrates argues that his questions didn't give any information to the boy that he didn't have already:[2]

Socrates What do you think, Meno. Has he answered with any opinions but his own?

Meno No, only with his own.

Socrates And yet he certainly didn't know, as we said a little while ago.

Meno What you say is true.

Socrates But he certainly had these opinions in him – or didn't he?

Meno Yes.

Socrates So someone who doesn't know something, whatever it may be he doesn't know, has true opinions in him about the very thing he doesn't know?

Meno It appears so.

And with that, Socrates is ready to drop the hammer:

Socrates And recovering knowledge which is within one for oneself is recollecting, isn't it?

Meno Yes indeed.

Socrates Well, the knowledge which this boy has now - he either acquired it sometime or else always had it, didn't he?

Meno Yes.

Socrates Then if he always had it, it follows that he was always in a state of knowledge. On the other hand, if he acquired it sometime, it could certainly not be in his present life that he has done so. Or has someone taught him geometry? For he will do just the same with anything in geometry or any other subject of knowledge. Has someone taught him everything, then? Presumably you should know, especially as he's been born and brought up in your home.

Meno No, I know that no one ever taught him.

Socrates And does he have these opinions or not?

Meno Apparently he must do, Socrates.

Socrates And if that is without acquiring them in his present life, doesn't it clearly follow that he had them and had learnt them at some other time?

Meno Apparently.

Socrates And that means the time when he was not a human being, doesn't it?

Meno Yes.

Meno is being very agreeable here, at a point where many of us might balk. But Socrates quickly explains a little more about what he means:

Socrates Well, if both during the time that he is a human being, and during the time that he is not, there are going to be true opinions within him which become knowledge when aroused by questioning, isn't his soul going to be for all time in a state of having learnt?

And that's his answer to where knowledge comes from: it resides within our souls, ready to be recollected when we encounter the right information.

In the same book Chomsky discussed the idea of being able to formally describe the regularities shared by human languages, noting that "The grammar of a particular language, then, is to be supplemented by a universal grammar that accommodates the creative aspect of language use and expresses the deep-seated regularities which, being universal, are omitted from the grammar itself."[3] He later began to use the term "Universal Grammar" to refer to the genetic endowment that supports language acquisition – the factors that constrain the set of languages we can acquire. Just as a grammar is the theory of a set of sentences, Universal Grammar is the theory of a set of *languages*, picking out those that are compatible with human minds.[4]

What form might such a Universal Grammar take? Chomsky entertained a variety of proposals over the years, refining his theory of what language is and what is needed to acquire it. However, one of these proposals – in vogue in the 1980s – is particularly intuitive: the idea that the syntax of human languages can be characterized by a set of principles that identify basic parameters that can vary across languages.[5] A specific language can then be identified by figuring out how to set those parameters. For example, in English we typically have to include the pronoun when we use a noun ("You know"), while in Spanish it is permissible to drop the pronoun ("Sabes"). All human languages can be assigned a value for this parameter – they either do or do not allow pronouns to be dropped.

This "Principles and Parameters" formulation of Universal Grammar strikes a nice balance between allowing the kind of variation we see in human languages while also making it easy to acquire those languages. If there were, say, 100 principles, each allowing 2 parameter values, we could specify 2^100 different languages – more languages than the number of stars in the universe. But a child encountering a new language need only resolve the values of those 100 parameters, which could potentially be done from a small amount of input. For example, a child need only hear a single sentence to discover that pronouns can be dropped.[6]

Proving language is unlearnable

The version of the poverty of the stimulus argument that Chomsky presented relied upon the intuitive complexity of a grammar, compared to the limited input that children receive. However, the argument would be much stronger if it were possible to definitively show that certain kinds of languages can't be learned from this input. Chomsky actually mentioned this possibility in his review of *Verbal Behavior*, writing "At the moment the question cannot be seriously posed, but in principle it may be possible to study the problem of determining what the built-in structure of an information-processing (hypothesis-forming) system must be to enable it to arrive at the grammar of a language from the available data in the available time."[7]

That possibility was realized only a few years later, when Mark Gold, a mathematician working at the RAND Corporation, proved a provocative theorem about the limits of learning.[8]

The details can get pretty hairy, so I'm going to introduce the basic ideas using a pedagogical device with as rich a history as the Socratic method: gambling.[9]

Imagine you meet a man who introduces himself as Dr. Gold. With a glint in his eye, he asks whether you might be interested in a wager. He proposes a simple game: He will choose a set of numbers. He will then say numbers that belong to that set out loud. If a number belongs to the set, he has to say it at some point – he isn't allowed to leave anything out – but he can repeat numbers as many times as he wants. Each time he says a number, you make a guess for which set he chose. If you eventually guess the right set, and *keep* guessing the right set even when he says additional numbers, you win. If you don't, you lose.

Gold says that to make it easy on you, for the first game he will choose one of only five sets. The first contains only the number 1, which I will write as $\{1\}$. The second contains 1 and 2, $\{1, 2\}$. The third contains 1, 2, and 3, $\{1, 2, 3\}$. And, of course, the fourth and fifth are $\{1, 2, 3, 4\}$ and $\{1, 2, 3, 4, 5\}$. We will call this **Game 1**.

So, should you take the bet?

The answer is yes. Even though you should be wary of any man with a glint in his eye, this is a safe bet. There is a simple strategy you can use to win the game: guess the set corresponding to the largest number Mark has said so far. At some point, he has to say the largest number in the set, and from that point onwards you will be correct.

Gold grudgingly hands over the cash. He turns to leave, but then turns back and asks whether you might be interested in a second wager. Double or nothing. The same game, just with different sets of numbers. How hard could it be?

In **Game 2**, the sets are just the same as before – $\{1\}$, $\{1, 2\}$, $\{1, 2, 3\}$ and so on – but this time, rather than the largest set containing five numbers, they keep going all the way to infinity. So the largest set is the one that contains all the natural numbers. Should you take the bet?

The answer is a definite no. There is no way to guarantee that you can win this game. The problem is that at some point you have to make a decision about whether to switch from guessing one of the finite sets – such as the set corresponding to the largest number Gold has said so far – to the infinite set containing all the natural numbers. If you're not willing to make the switch, then you'll lose the game if Gold has chosen the set of all natural numbers. If you do make the switch, then you lose the game if Gold has chosen some finite set.

What does all this have to do with learning language? Well, even though we have been talking about sets of numbers, remember Chomsky had asserted that a language is a set of sentences. So we can apply the same kind of analysis to languages, where the numbers correspond

to possible sentences. For one example of a simple mapping from numbers to sentences, you could imagine that the number represents the number of times you can repeat the word "hairy" in the finite state grammar from Chapter 5. (I warned you that the details can get pretty hairy.)

Substituting sentences for numbers, Gold's game becomes a way of characterizing the problem of learning a language: guessing what set of sentences is grammatical after hearing sentences from the language. A collection of languages is learnable if there is a strategy that always succeeds in identifying the correct language from examples – if you can guarantee you can win Gold's game. So, for numbers, the sets in Game 1 are learnable, but the sets in Game 2 are not.

Characterizing learning in this way, Gold proved a remarkable theorem: no collection of languages that includes all finite languages and at least one infinite language is learnable. The proof was basically the argument I gave for why you can't win Game 2. Finite languages are just finite sets of sentences, the equivalent of the sets of numbers in Game 1. They can be generated by finite state grammars, meaning that they are a subset of the finite state languages. So, adding one finite state language that generates infinitely many sentences – like the hairy dogs language – to all finite languages creates an impossible situation where you can't know whether the language you are hearing is finite or infinite. It only becomes worse if you add more finite state languages. Gold had thus shown that, under his definition of learning, even learning which finite state language somebody is using is impossible, let alone context free or context sensitive languages.

Gold's theorem provided mathematical force to the poverty of the stimulus argument. By showing that learners cannot distinguish between a relatively simple collection of languages, it supported Chomsky's idea that some kind of constraints on the languages considered by learners are necessary. Other researchers showed that Gold's result still holds if we make changes to some of his assumptions – if learners find out that some sentences do not belong to the language, or if their guesses do not need to be exactly right. These results were taken as a guide for investigating what children actually hear and how they use it in learning language.

So far we have seen some collections of languages that are unlearnable, by Gold's criteria. But what makes a collection of languages learnable? The answer to this question was discovered by Dana Angluin, a computer scientist at Yale.[10] To illustrate her results, we are going to play three more games following the same rules as Gold's game. For each one, think about whether you would be willing to play.

Game 3 The sets are the numbers $\{1, 2, 3, 4, 5\}$, but each missing a number. So the first set is $\{2, 3, 4, 5\}$, the second is $\{1, 3, 4, 5\}$, and so on.

It shouldn't come as a surprise that this is a safe bet. The winning strategy is to guess the set corresponding to the smallest number you have not seen so far.

Game 4 The sets are the natural numbers $\{1, 2, 3, \ldots\}$, but each missing a number. So the first set is $\{2, 3, \ldots\}$, the second is $\{1, 3, \ldots\}$, and so on, going all the way to infinity.

You might be a little scared by infinity showing up again, but this game is actually winnable. The strategy from Game 3 is also a winning strategy for this game – at some point you will hit on the right hypothesis and be correct from then on.

Game 5 The sets are the natural numbers $\{1, 2, 3, \ldots\}$, but each missing a number. So the first set is $\{2, 3, \ldots\}$, the second is $\{1, 3, \ldots\}$, and so on, going all the way to infinity. However, we also include the set of all natural numbers.

You should not play this game. Including the additional set of all natural numbers creates the same problem of having to decide whether to switch to this set or not, so there's no way to guarantee that you will win.

Game 4 illustrates that it's possible to have an infinite collection of languages and still learn the right language. Game 5 shows that the critical thing is the relationship between languages. Angluin proved that a collection of languages is learnable provided no language is a subset of any other.[11] It is the fact that some languages are subsets of others that creates the challenge for learners that Gold identified.

Chapter 5

No free lunch

Hume's 18th century skepticism has an echo in 20th century theoretical analyses of machine learning. When exploring what the limits of learning systems might be, David Wolpert and William Macready proved a set of "no free lunch" theorems: in the absence of other assumptions about the task being performed, no learning algorithm is better than any other.[12] To get an intuition for this result, consider a simple learning problem: you will get to observe the values of two variables (call them x_1 and x_2) and have the opportunity to predict the value of a third (x_3). Assume the variables can only take on the values 1 and 0, and you observe that both x_1 and x_2 take the value 1. What do you predict for x_3?

If you predicted that x_3 would also take the value 1, then that seems very reasonable. You could arrive at this prediction via a variety of different algorithms such as predicting the most common value of the previous observations, or the average. But if all possible worlds are equally likely, these algorithms will work just as well as algorithms that do the complete opposite — taking the least common value, or the average of the values you didn't observe. To see this, it is helpful to make a table of all of the possibilities, similar to the truth tables from Chapter 2:

x_1	x_2	x_3
1	1	1
1	1	0
1	0	1
1	0	0
0	1	1
0	1	0
0	0	1
0	0	0

If you just look at the rows where x_1 and x_2 take the value 1, you can see that x_3 is 1 in one row and 0 in the other. So if all these worlds are equally likely, you will get the right answer 50% of the time. In fact, it doesn't matter what procedure you use for making a decision, the best you can do is 50% correct.

How can you do better than this? Well, if the future is like the past, then that places a constraint on possible worlds. We can implement this by removing the worlds where x_1 and x_2 are the same but x_3 is different, as these are cases where the future (x_3) differs from the past (x_1 and x_2). This gives us just six possible worlds:

x_1	x_2	x_3
1	1	1
1	0	1
1	0	0
0	1	1
0	1	0
0	0	0

Now, predicting that x_3 will have the most common value taken by x_1 and x_2 works well – it produces the right answer in 4 out of 6 cases. By contrast, predicting the least common value works poorly, producing the right answer in only 2 out of 6 cases. We have made it possible to have a reasonable strategy for inductive inference by making the assumption that the future will be like the past. By extension, the only reason that any machine learning algorithm works better than any other is that the world we live in is one that has regularities that align with those that are detected by the algorithm: it's not that any particular algorithm is intrinsically better, it's that some algorithms are better aligned with the structure of the world that we live in.

Notes

1. The results presented in this paragraph appear in Chomsky and Schützenberger, "The algebraic theory of context-free languages."

2. The translation of the *Meno* used here is from Day, *Plato's Meno in focus.* The quoted sections are 85b-86b, pp. 55-56.

3. Chomsky, *Aspects of the theory of syntax*, p. 7.

4. See, for example, Chomsky, "Reflections on language," p. 29.

5. See Chomsky, *Lectures on Government and Binding.*

6. For an analysis of this kind of model of language acquisition, see Gibson and Wexler, "Triggers."

7. Chomsky, "A review of B.F Skinner's Verbal Behavior," p. 58.

8. Gold, "Language identification in the limit."

9. The presentation here is inspired by Osherson, Stob, and Weinstein, *Systems that learn: An introduction to learning theory for cognitive and computer scientists.*

10. Angluin, "Inductive inference of formal languages from positive data."

11. Technically, she showed that the relevant criterion is that each language L has a finite subset T such that for all other languages L', if $T \subseteq L'$ then $L' \not\subset L$, for which no language being a subset of any other is a sufficient condition.

12. For some initial results in this literature, see Wolpert and Macready, "No free lunch theorems for optimization."

References

Angluin, Dana. "Inductive inference of formal languages from positive data." *Information and control* 45, no. 2 (1980): 117–135.

Chomsky, N. "A review of B.F Skinner's Verbal Behavior." *Language* 31 (1959): 26–58.

———. *Aspects of the theory of syntax*. Cambridge, MA: MIT Press, 1965.

Chomsky, Noam. *Lectures on Government and Binding*. Mouton de Gruyter, 1981.

———. "Reflections on language" (1975).

Chomsky, Noam, and Marcel P Schützenberger. "The algebraic theory of context-free languages." In *Studies in Logic and the Foundations of Mathematics*, 26:118–161. Elsevier, 1959.

Day, Jane M. *Plato's Meno in focus*. Routledge, 2003.

Gibson, Edward, and Kenneth Wexler. "Triggers." *Linguistic Inquiry* 25 (1994): 355–407.

Gold, E. Mark. "Language identification in the limit." *Information and Control* 10 (1967): 447–474.

Osherson, Daniel N, Michael Stob, and Scott Weinstein. *Systems that learn: An introduction to learning theory for cognitive and computer scientists.* The MIT Press, 1986.

Wolpert, David H, and William G Macready. "No free lunch theorems for optimization." *IEEE Transactions on Evolutionary Computation* 1, no. 1 (1997): 67–82.