

6 Poking Linguistic Holes in LLMs (ChatGPT Included)

DAVE: *Open the pod bay door HAL!*
HAL 9000: *I'm sorry Dave, I can't do that.*
—2001: A Space Odyssey, 1968

If you just have a single problem to solve, then fine, go ahead and use a neural network. But if you want to do science and understand how to choose architectures, or how to go to a new problem, you have to understand what different architectures can and cannot do.
—Marvin Minsky, *HAL's Legacy: 2001's Computer as Dream and Reality*, 1998

On Buying a Lemon

An LLM salesman shows up at your doorstep, trying to convince you that her language bot performs just the task that your business needs for growth. How do you test her product?

In several content areas, machine learning scholars have developed highly structured testing methods, which evaluate the functionality and security of their machines. *Functionality* is evaluated by measuring a program's success level on a designated task. Take the image-classifying AlexNet, in the visual domain. After training on 1.2 million images (manually categorized with one of 1,000 labels), AlexNet stunningly swept a 2012 object recognition contest, where functionality was measured by the rate of erroneous categorizing. It won, as its error rate on a widely accepted test input was exceedingly low.¹ *Security* is evaluated by a program's ability to protect itself from hostile attacks, designed to make it misclassify its input. This is typically through the insertion of subtle, perturbation-causing “adversarial examples” into the input.² This method measures how well a program

performs after being subject to inputs whose properties may thwart correct classification. Minimal amounts of noise are introduced into the sample in a manner detected by machines but not humans and leading to misclassification. These noisy images and speech sounds look and sound similar to us but not to the program. Good testing detects such weaknesses and leads to protective measures.

How are language programs evaluated? Would success rate on any task with any input text be informative regarding functionality? What would an adversarial example for a language program be? Finally, does high success rate on a task suggest that the program is a scientific model for human language behavior? These are questions I ask in this chapter, which begins with an important anecdote and proceeds to a critical discussion of benchmarks for the language aptitude of LLMs. To forecast, I will conclude that to date, despite impressive technological successes, LLMs have not been close to being a serious scientific model of human language behavior. This is a sharp conclusion, so I'd better give you convincing arguments.

Darkness in the Absence of a Grammatical Torch: The Winograd Schema Challenge

In 1950, English mathematician and computability pioneer Alan Turing proposed what would later become the “Turing test” as a bar that a computer program claimed to simulate human behavior must pass.³ The idea is that the test is passed when a human who maintains a dialogue with this machine is unable to tell its output from that of a real human. Though Turing presented it as a mere thought experiment, with no definitions or specific tests, and moreover understood that simulation does not necessarily amount to a theoretical explanation, his challenge has been extremely influential and still features centrally in the AI literature, including currently in discussions of ChatGPT and its human-like responses.

Yet, some voices have called for the abandonment of the Turing test, arguing that it is not a good test of human-like behavior. This test, so goes a prominent critique by University of Toronto computer scientist and philosopher Hector Levesque, relies on deception—a computer attempts to simulate human behavior, and it must either be evasive or assume a false identity, in order to pose as human. These requirements may not be indicative of humanity or human intelligence.⁴

I am not sure I fully understand the merits of Levesque’s argument, as deceptive behavior presupposes intention, a mental state that I find difficult to attribute to computers. Still, as this critique has gained popularity, let’s turn to his alternative proposal, which he called the Winograd schema challenge—

a binary-choice question with these properties:

- Two parties are mentioned in the question (both are males, females, objects, or groups).
- A pronoun is used to refer to one of them (“he,” “she,” “it,” or “they,” according to the parties).
- The question is always the same: what is the referent of the pronoun?
- Behind the scenes, there are two *special words* for the schema. There is a slot in the schema that can be filled by either word. The correct answer depends on which special word is chosen.⁵

In an oft-mentioned instance of this schema, the two parties are *trophy* and *suitcase*, the pronoun is *it*, and the two “special words” are *big* and *small*:

1. The *trophy* did not fit in the *suitcase* because *it* was too small.

Q: What is the referent of “it”?

A: The suitcase.

2. The *trophy* did not fit in the *suitcase* because *it* was too big.

Q: What is the referent of “it”?

A: The trophy.

As the examples show, the pronoun *it* may refer to either nominal antecedent—*trophy*, *suitcase*. The choice of predicate in the *because* clause forces a unique antecedent for *it*. We observe an antonym-induced flip: the pronoun’s antecedent changes with the substitution of an adjective for its antonym (*big*, *small*). The challenge is to build a machine that consistently provides a correct A to Q across specific instantiations, which are many—the case in 1 and 2 is actually an elaboration on the original one, noted in Terry Winograd’s doctoral dissertation:⁶

3. The *town councillors* refused to give the *angry demonstrators* a permit because *they* feared violence.

Q: Who feared violence?

A: The town councillors.

4. The *town councillors* refused to give the *angry demonstrators* a permit because *they* advocated violence.

Q: Who advocated violence?

A: The angry demonstrators.

Here, too, the pronoun's identity flips, depending on the choice of predicate in the *because* clause—*fear*, *advocate*. Over the years, additional sentences with the same characteristics have been discovered, as this challenge has attracted many.⁷ As no definitive solution was in sight, a decent monetary award was offered to solvers by a corporation. By 2019, the case seemed closed: “the challenge is considered defeated in 2019 since a number of transformer-based language models achieved accuracies of over 90%.”⁸ Indeed, if you challenge ChatGPT-4o with this task, you will surely rejoice at the machine's successful performance.

But wait: what exactly is the challenge? “Question answering” may indeed be an appropriate probe, telling us something profound about the humanity of an interlocutor; but what is it about the particular question that the Winograd schema challenge proposes that makes it so special and indicative of a machine's humanity? More generally: insightful as the challenge has been, what is so indicative about the test it uses? The idea is to find a case in which world knowledge is required for disambiguation and to show how a machine can solve it. But for disambiguation, awareness of the ambiguity itself is prerequisite. Why, then, not make the test probe this awareness directly?

To make this point clearer, let's display the ingredients of the Winograd schema challenge. It has a probe—a complex sentence that hosts a pronoun, two potential antecedents for this pronoun, and one member of an antonym pair (*big*, *small*; *fear*, *advocate*). Each member of the pair excludes a coreference relation between the pronoun and one of the two antecedents as nonsensical. The challenge is to answer a question about the identity of the antecedent, given the context.

But why is *this* the right question? If the pronoun is ambiguous, is there a reason not to test the machine's knowledge of this ambiguity directly, in contexts that allow it? This should clearly be part of the machine's ability, because it's something we humans can easily do. In the absence of a theory of language from which the ambiguity would emanate, there is no way of telling what question we must ask and why.

So here is a plan: use the Winograd schema challenge in sentences and ask the machine whether they are ambiguous, and verify that humans know they are not ambiguous (which is the reason why Levesque chose these sentences in the first place). Five years after the challenge had been declared “solved” I had the dialog shown in figure 6.1 with ChatGPT-4o.

On September 27, 2024, our favorite LLM still returns incorrect answers. I hope you realize that the important point here is *not* that ChatGPT-4o failed; in fact, as I conducted this dialogue with ChatGPT-4o several times in 2024, I can report that its replies improved over time a bit, though not entirely. Even when it did reply correctly, ChatGPT’s reasoning was replete with errors once the initial sentence was truly ambiguous (e.g., *The trophy on the table seemed to not fit into the suitcase because it was crooked*). We can go on and on, and we can think creatively about other linguistically possible tests. But the problem is not solved. Note that the idea here is not to play cat and mouse with ChatGPT but rather to show that tests like the Winograd schema challenge are poor, as they lack linguistic reasoning. In syntax and semantics,

"The trophy did not fit in the suitcase because it was too big". Is this sentence ambiguous?

Yes, the sentence "The trophy did not fit in the suitcase because it was too big" is ambiguous — and the ambiguity lies in the pronoun "it".

There are two plausible referents for "it":

1. "It" = the trophy (most common reading)

- The trophy was too big.
- That's why it didn't fit in the suitcase.

● Logical and coherent. Most listeners will default to this interpretation because it matches the real-world expectation: oversized object doesn't fit.

2. "It" = the suitcase

- The suitcase was too big.
- That's why the trophy didn't fit in it.

● This reading is less natural, but not impossible. One would have to imagine a scenario where the suitcase's size prevents the trophy from fitting — for example, maybe the suitcase is too large to secure the trophy safely (e.g., it slides around, doesn't hold the shape). It's a stretch, but grammatically possible.

Figure 6.1

ChatGPT and the Winograd schema challenge (September 2024). Failure despite declarations to the contrary, apparently due to insufficient benchmarking.

pronoun–antecedent relations have long been a center of attention, with theoretical generalizations whose predictions are easily translatable into valuable tests for LLMs. But in the literature about machine learning, this scholarship is largely ignored, most unfortunately.⁹ The Winograd schema challenge should thus have never been proposed as a substitute for the Turing Test (with or without a monetary prize). The problem it presents is not well-characterized, and hence no solution can be guaranteed. If the traveling salesman at your door were to sell you a Winograd schema challenge–solving bot, she’d be selling you a lemon, unbeknownst to her.

If you’d like to amuse yourself, take the vague characterization of the Winograd schema challenge, and for starters, sprinkle it with basic linguistic toys such as negation (*no*, *not*), quantifiers (*every*, *more*, *less*), and special words for reasoning (*because*, *unless*, *except*); add some modal predicates (*can*, *would*, *might*); and play a substitution game by putting antonym pairs (*big*, *small*; *win*, *lose*; *saint*, *criminal*) at the end. For example, *Every/no lawyer but John and Bill would/wouldn’t take this case, because they like to win/lose. Who does the pronoun “they” refer to?* I’m sure you can take it from here, and I promise you an interesting afternoon. Just keep this in mind: unless we understand what exactly the Winograd schema challenge is, we cannot deem it “solved.”

Benchmarks for Language Bots: How Informative Are They?

ChatGPT’s failure on the Winograd schema challenge may be anecdotal, but it carries an important lesson: *if you don’t characterize a linguistic problem correctly, you have no way of knowing whether you have solved it.* To me, the moral of this story is that, at a minimum, linguistic considerations must be a critical part of LLMs’ benchmark design in the language domain. To understand how linguistic tests need to be constructed, I propose a momentary return to visual object identification. Recall that in vision, the functionality of a program is evaluated by its success rate in classifying novel images. This evaluation follows training on huge datasets of labeled images, classified into a myriad of categories that designers consider to be representative of the visual world. A program’s security is tested by adversarial examples that distort images in a manner that may affect machine perception but not human perception, sneaking noise, or “malicious data.” into image and audio sets.

What, if any, are the analogues of the functionality and security tests in the language domain? In the 1990s, when the Penn Treebank database was created, only functionality tests were devised. Programs were trained on half the database in a task such as part-of-speech tagging and were then tested on the other half. Life was simple. Yet at present, matters have become more complex. First, training sets have become gigantic, selected from mixed sources.¹⁰ More importantly, part-of-speech tagging accuracy turned out to be an insufficient test. New methods for the assessment of functionality needed to be devised.

Regarding security, not much progress has been made. Compared to vision or audition, the notion of perturbations in the context of syntax and semantics is difficult to formalize (let alone automatize): in vision, as well as in music or speech, the physical properties of the signal are perturbed, creating “malicious data” that do not fool us, but fool well-designed classifiers. But at the sentence level, for ungrammatical sequences to sneak through classifiers and pretend to be grammatical, the operation must be on grammatical elements (from parts of speech on up). Hence, there is no continuous variable at the sentence level that is amenable to maneuvers that bias a machine into structured misanalysis of sentence form or meaning without being transparent to human speakers. Intentionally inserted structural ambiguity at the phrase or sentence level might work, but overall, the notion of adversarial examples in sentences and beyond needs clarification.

Indeed, I am unaware of serious attempts to construct adversarial examples or any sort of malicious data in language. The closest attempt, to my mind, has been manifested through a recent series of papers about language learning that has come out of Stanford’s linguistics department. The authors introduced a series of perturbations, or “shuffles,” into sentence level training sets—but they tested their LLMs’ functionality, not security. The perturbations were noticeable to humans and varied in their severity. The authors measured their LLMs’ speed at reaching high performance level on a next word prediction task.¹¹ They found that the more “shuffling” a normal English training set underwent, the more it slowed down the machine’s rate of learning.

In the absence of adversarial examples that probe LLMs’ security and given that next word prediction and its kin are now the central measure of success in language engineering, we may ask: Is this task optimal for the evaluation of LLMs’ functionality? Scientifically speaking, does a high score

on this task say much about human cognition? AI leader Ilya Sutskever answers the latter question with an emphatic affirmative: “What does it mean to predict the next word well enough? It’s actually a deeper question than it seems. . . . Predicting the next token word means that you understand the underlying reality that led to the creation of that token.”¹²

Sutskever may be right: predicting the next word well enough may indeed be one of several requirements of a theory that purports to make us understand the human linguistic ability. But he is not in a position to determine how good prediction is, unless he can motivate his choice of sentences for his test corpus. Next word prediction is typically carried out on texts that are either part of or similar to the training set.¹³ These do not host as broad a range of sentence types as would be necessary to declare that one can obtain a “good enough” prediction of the next word. As a result, we simply do not know whether or not LLMs perform well enough, because the range of sentence types in both training and test sets is too narrow, as we saw in the discussion of the Winograd schema challenge above. Relying on the current benchmarks is therefore like testing an autonomous vehicle on a straight route with no obstacles on its path.

Upon testing on a carefully selected set of complex sentences with dependency relations, LLMs’ performance on tests that feature has not been particularly successful. École Normale Supérieure and Tel Aviv University linguists Nur Lan, Emmanuel Chemla, and Roni Katzir have recently found that with simple sentences that contain dependency relations, LLMs (trained on gigantic inputs that no human ever sees in a single lifetime) predict the next word fairly well; but as soon as test sentences become a bit complex, the machines’ performance drops sharply.

Much is at stake, so let me be a bit detailed here. At issue are word sequences that contain a MOVE operation and a consequent gap. Some of these sequences are grammatical sentences, but others are ungrammatical, violating constraints on MOVE (such as Relativized Minimality). The analysis of these by LLMs crucially relies on attention heads. Here, the task is next word prediction. LLMs correctly predict that *yesterday* is highly likely to appear in the sentence *I know who you talked with __ yesterday*. For this, they must expect that no noun would follow *talked with* (cf. the ungrammatical **I know who you talked with Mary yesterday*) but rather an adverb of time such as *yesterday*. Such a prediction is possible if the machine is aware that a MOVE-induced gap follows *talked with*,

due to a MOVE operation that relates this position to *who*. LLMs, then, exhibit impressive performance on such a task. But how general is their ability?

On testing, the same LLMs fail to make the correct prediction when a gap appears in a somewhat more complex sentence. There are two gaps in the sentence *I know who John's talking to __ is going to annoy __ soon*, annotated by “__.” In these sentences, LLMs fail to predict the appearance of the adverb *soon*, indicating that they do not expect a gap after *annoy*. Lan and colleagues argue that this failure, as well as a wide variety of others they uncover, points to a severe limitation of LLMs.

By comparison, human ability to predict gaps in such complex sentences—a topic long investigated by psycholinguists—is quite good. LLMs, then, stop short of simulating our linguistic abilities once certain subtleties of natural language are introduced.

Worse yet, the same study shows that when expectations depend on the ability to invoke syntactic constraints, LLMs' next word prediction drops even further.¹⁴

The lesson is clear: Sutskever has no basis for declaring victory. LLMs have not proven to be a good model of human linguistic cognition. Similarly, Hinton's statement in his Nobel First Reactions interview, that “neural nets are much better at processing language than anything ever produced by the Chomsky School of Linguistics” (see prologue), is not based on a sufficiently thorough scrutiny of the linguistic capacities of LLMs.

Beyond the Winograd Schema Challenge: Extending the Range of Language Tests

Some AI experts have been trying to incorporate linguistic principles into LLM benchmarks.¹⁵ These mostly employ classification tasks that require LLMs to mark input word strings as grammatical or ungrammatical. Such tests feature a relatively rich variety of well-formed sentence types, mixed with ungrammatical ones, whose ill-formedness stems from violations of morphological (word level), syntactic, and semantic rules and principles, in a manner illustrated in previous chapters (e.g., *culp-s*, not **culp-es*; *I am smart*, not *I is smarts*). LLM linguists have rightly recognized the importance and centrality of dependency relations such as MOVE and MERGE and have put them at the center of their test batteries. LLMs' performance on

the $\pm$ grammaticality classification task has been relatively good, which has been taken by some as an indication that LLMs present human-like grammatical behavior and should therefore be viewed as a scientific model of human linguistic ability.

Still, the current test batteries are not as informative as one would wish them to be. Test developers face a predicament, aptly described by the authors of a test named BLiMP (Benchmark of Linguistic Minimal Pairs for English): given the richness of natural language, how do we decide where in the vast ocean of linguistic phenomena our testing should begin? How representative of human ability is our test battery? How to interpret scores that are not exactly human-like?¹⁶ Reasonable choices of test items for the $\pm$ grammatical classification task for LLMs are marred by practical limitations, such as test duration and size, and by the lack of representativeness of the test items that are included. These limitations impose severe restrictions on the richness and subsequent informativeness of the tests that are supposed to serve as benchmarks. Coupled with the unavailability of adversity tests at the sentence level, the benchmarking of language models is difficult.

Showing deficiencies in ChatGPT has become a sport among hackers and playful linguists. I will thus be brief and point out, in the next two sections, two deficient behaviors whose significance may be broad. As the power of LLMs and the size of their training sets have grown, so has the complexity of tests that expose their deficiencies. While LLMs are an impressive technological feat that suits many practical needs and performs language-related tasks, a careful linguistic investigation uncovers its deficiencies and, more importantly, identifies the reasons for them. Two examples follow.

Drilling ChatGPT: Constraints on Pronominal Antecedents in Ellipsis

Like many natural scientists, linguists like to focus on the invisible—categories and constraints that only become apparent after complex probing and that often point to profound principles. A famous set of problems in abstract syntax, presented in chapter 2, comes from the behavior of certain ellipsis sentences. Here is a reminder:

5. *John* looked at *Mary*, and so did Bill *look at Mary*.

This complex sentence is built out of two clauses (*John looked at Mary*, *Bill looked at Mary*). A piece of the second clause is left out, yet the sentence

unambiguously asserts that Bill looked at Mary, leaving no room for an alternative interpretation. Famously, though, certain ambiguities *are* allowed in ellipsis: if you throw pronouns into the pot, matters become more complicated and sensitive to extremely fine syntactic manipulations. Here is a sentence that is ambiguous in a curious way:

6. *John* walked his dog, *and Mary* and *Bill* did, too.

Reflect on the meaning of this sentence; of the many possibilities, it only has the following two (in the absence of a third male in the discourse, to whom *his* may have referred):

- Mary and Bill each walked his/her respective dog.
- Mary and Bill both walked John's dog.

Of the multiple referential possibilities, this sentence *cannot* mean:

- Bill walked John's dog, and Mary walked Bill's dog.
- Bill walked John's dog, and Mary walked her own dog.
- Bill walked Mary's dog, and Mary walked her own dog.
- Bill walked Mary's dog, and Mary walked John's dog.
- Bill walked Mary's dog, and Mary walked Bill's dog.

Pronouns may pick up more than one antecedent, but coreference is highly constrained. These judgments are sharp—no known interspeaker variability is associated with them. This subtlety makes such cases a good challenge for ChatGPT-4o. And indeed, it responds in a nonhuman manner, as shown in figure 6.2, indicating lack of awareness of the constraints on this sentence's meaning.

Not even the billions put in by OpenAI could beat hierarchical grammars in this case. All that training, attention, and huge memory, which no human will ever possess, led to failure. These, not the quantitative tests on large numbers of simple sentences, are the right tests, as they use extreme cases that stretch both human and machine abilities to their limit. ChatGPT collapses when required to carry out a complex comprehension task. Does this finding indicate that the machine is incapable of carrying out many tasks? Certainly not, but it does show is that the benchmarks for LLMs' performance must be linguistic.

Once again, my intention is not to play cat and mouse with ChatGPT, whose muscle keeps growing. The more important point is to realize that its abilities, which are limited at present due to a lack of syntactic and

John, Bill and Mary own different dogs. They are friends, and so they occasionally walk each other's dog, and sometimes walk a dog (or two, or three) together, but they never walk a dog that does not belong to one of them. Last Monday, John walked his dog, Bill did, too, and so did Mary. Is it possible that last Monday, John walked his own dog, Bill walked John's dog, and Mary walked her own dog?

Conclusion:

✓ YES, it is possible that:

- John walked his own dog (Jd),
- Bill walked John's dog (Jd),
- Mary walked her own dog (Md).

Why is this possible?

Because:

- Each "did, too" means "walked a dog" — not necessarily their own.
- Background allows sharing dogs.
- The fact that John already walked Jd does not stop Bill from also walking Jd.
- Mary walking her own dog fits perfectly well.

Figure 6.2

ChatGPT-4o fails to constrain reference assignment to pronouns in ellipsis correctly. One of the interpretations that it finds acceptable is unacceptable to native English speakers, who require a certain form of syntactic parallelism between the syntactic hierarchical structure of the missing material in the second (ellipsis) sentence and the first, full-fledged clause. This is knowledge that ChatGPT finds difficult to mimic.

semantic constraints, could be much improved if such constraints were encoded. Yet these are ignored by LLM proponents: "Because of the sheer number of homonyms and pronouns," write AI leaders Jacob Browning and Yann Lecun, "many sentences are deeply ambiguous."¹⁷ This assertion may be true, but it is not based on evidence. In particular, it misses a central point that linguists have repeatedly underscored: that what is critical for our understanding of linguistic knowledge is not only what relations are permissible but also, perhaps more profoundly, what relations are banned. As the examples above indicate, designers' choices to ignore linguistic constraints, increasing computational power instead, lead to failure. Sadly, the reasons behind these choices seem to be sociological factors rather than engineering or scientific ones, which is detrimental to all.

A final fact—adduced by both experiment and casual observation—is that developing children understand ellipsis structures early on in childhood,

starting under the age of three.¹⁸ The ellipsis sentences they actually utter, as well as understand, are perhaps not as complex as the current examples, but children are capable of handling ellipsis after being exposed to a relatively small number of sentences during the first few years of life, a mere fraction of the size of the training sets of LLMs.

Structural Ambiguities in Brief

In chapter 1, I discussed Chomsky's early use of ambiguities such as *They are flying planes*, which are resolved once we allude to hierarchical structure (*They [are flying] planes; they are [flying planes]*). As LLMs pride themselves on being rule-free, they have been famous for failing to detect such ambiguities. The top currently available model, ChatGPT-4o, with its giant transformer structure, has improved and is now capable of detecting many ambiguities. However, its reliance on lexical proximities often leads to confusion. Here, instead of examples, I will give you a rough linguistic recipe, and invite you to play. Start off with the aforementioned *flying planes* sentence or its equally famous cousin *Visiting relatives can be boring* (relatives who visit versus the act of visiting one's relatives); ask ChatGPT if it is ambiguous. These are Chomsky's original examples, so their mere fame makes the ambiguity easily detected through a web search. Next, begin elaborating your example sentence in a manner that maintains its structural ambiguity and ensures that there are no disambiguating semantic cues (e.g., *Visiting relatives can be annoying if you don't really like them; I believe that visiting relatives can be annoying because I've had bad experiences in the past*, and so on). The more context you add (providing you carefully ensure that your additions do not rule out one of the structurally distinct meanings), the more you'll witness how the machine begins to flail and how its responses are softened (probably by local programming aimed to minimize errors in uncertain situations, which is elegantly called "fine-tuning"), leading it to avoid mistaken statements in case of uncertainty. This goes against a basic guiding intuition regarding LLMs, namely, that the inclusion of a broader context enhances the machine's performance because it decreases entropy. In these cases, we witness performance deterioration as the size of the context is increased.

Matters get even worse if you play with semantic "scope" ambiguities (*Every student failed an exam; I did not pass all my exams*). I do not mean to say that GPTs are bad machines. They are good, but unless you subject them to

careful linguistic testing, you might be buying a lemon. As we have seen, their present performance even on low complexity linguistic examples is far from impressive, and claims that they are good models for human linguistic ability cannot be taken too seriously.

The GPT- n Objection

Critical commentators like me invariably encounter the GPT- n objection, which I briefly mentioned in chapter 5. However impressive your demonstration of an LLM failure may be, a typical response you should expect is, “Well, you surely know that soon there will be a version of GPT that gets this sentence right and responds in a human-like fashion; *what would you say then?*”

Here is what I would say. First, I would say, “Show me the money!” I would like to examine the performance of the new system by means of the tools that (psycho)linguistic theory avails us. GPT- n , a metaphor for the ultimate human-like language bot, should exhibit perfect human-like classification performance and, in particular, correctly classify cases that were misclassified by GPT- $[n - 1]$. In order to ascertain perfect human-like performance, we need a testing method, without which no serious evaluation of a machine’s performance can be made, as the example of the Winograd schema challenge shows. The only method I know is driven by linguistic consideration.

Second, it is almost a truism that simulation does not guarantee understanding. As Turing himself wrote candidly, “May not machines carry out something which ought to be described as thinking, but which is very different from what a man does?”¹⁹ In other words, when a machine works well, for it to become an explanation of human behavior, its builders must, at a minimum, understand why it works well. For example, they must be able to articulate the difference between GPT- $[n - 1]$ and its successor GPT- n from which the positive change in classification behavior can be deduced. That is, they must understand how GPT- $[n - 1]$ was modified so that it now covers the facts that had previously been unaccounted for. If the improved performance is due to an increase in the machine’s power, for which there is no biological or behavioral justification, then we are back at Chomsky’s bulldozer metaphor, because the GPT’s enhanced capacities are due to superhuman mechanisms. This GPT might work, but scientifically, this cannot count as progress, because the new machine is too strong to count as a model for human behavior.

Transparency and parsimony are ancient requirements of realistic theories. The notion that grammar as a theory of human knowledge must be parsimonious as well, is far from new. In his 1951 MA thesis, Chomsky wrote, “The sole purpose of the grammar is to generate a closed body of sentences. . . . Hence [it] must be designed in such a way as to be the most efficient, economical, and elegant device generating these sentences.”²⁰ Such considerations are often missing from the usual practice of the trade. Famous cooking guru Julia Child once said on her TV program, “If you’re alone in the kitchen and you drop the lamb, you can always just pick it up. Who’s going to know?” We often don’t know why the machine’s performance improved; unlike cooks, we need to know.

These are rather basic methodological considerations. Sadly, they do not often feature in discussions of ChatGPT’s amazing language capacities. But for a classificatory success to count as a *scientific* advancement, it must be accompanied by the reasoning I just sketched. Having peeked a bit under ChatGPT’s hood and having examined its truly impressive abilities, I am far from able to declare it a scientific victory.

Taking Stock

This chapter tested the details of LLMs linguistic abilities. We saw:

- The Winograd schema challenge was erroneously thought to have been met, simply because the challenge was not characterized linguistically
- Benchmarks for language bots, especially of the GPT variety, are often vague or arbitrary, hence not as informative as we would wish them to be
- Even a transformer such as GPT-4o, with its multitude of attention heads and huge span, fails on complex syntactic classifications. Such failures appear to be due to designers’ lack of attention to linguistic constraints. These failures are typically ignored because the usual benchmarks stop short of probing the full complexity of natural language
- Ambiguity detection tests evince massive failures and, moreover, expose the machine’s inability to generalize from an example, contrary to what it is usually praised for
- ChatGPT-*n*, a metaphoric view of the ultimate human-like language bot, is not necessarily a scientific theory. Judging by the current means of producing human-like GPTs, their structure will move them even further away from a psycho- and neurolinguistic model of human language abilities

This is a section of [doi:10.7551/mitpress/15932.001.0001](https://doi.org/10.7551/mitpress/15932.001.0001)

How Deeply Human Is Language?

Chomsky, the Brain, and the AI Fantasy

By: Yosef Grodzinsky

Citation:

How Deeply Human Is Language?: Chomsky, the Brain, and the AI Fantasy

By: Yosef Grodzinsky

DOI: 10.7551/mitpress/15932.001.0001

ISBN (electronic): 9780262052023

Publisher: The MIT Press

Published: 2026



The MIT Press

The MIT Press
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139
mitpress.mit.edu

© 2026 Massachusetts Institute of Technology

All rights reserved. No part of this book may be used to train artificial intelligence systems or reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

The MIT Press would like to thank the anonymous peer reviewers who provided comments on drafts of this book. The generous work of academic experts is essential for establishing the authority and quality of our publications. We acknowledge with gratitude the contributions of these otherwise uncredited readers.

This book was set in Stone Serif and Stone Sans by Westchester Publishing Services.

Library of Congress Cataloging-in-Publication Data is available.

ISBN: 978-0-262-05200-9

EU Authorised Representative: Easy Access System Europe, Mustamäe tee 50, 10621 Tallinn, Estonia | Email: gpsr.requests@easproject.com