Searle, Bender and Koller, syntax, semantics

Large language models generate language in a peculiar manner. Although they perform well in specific tasks such as coding and creating parodies, they are prone to making unusual errors. Since they are essentially designed to predict the next word, they can be presented with unrealistic scenarios and still predict how to complete them. For example, when tasked with developing the Seinfeld Streep theorem, they may concoct a mathematical formula, as they are trained with text containing theorem names and math formulas. Although they can generate plausible continuations, they may produce false or fictitious text. This phenomenon is of significant interest to software engineers and philosophers alike, as LLMs may play an essential role in our lives, but it is crucial for them not to invent things. For philosophers, this creates a unique perspective on the nature of meaning and the fundamental concepts of linguistics.

In particular, the strange abilities of today's LLMs offers a novel perspective on the old question, made most famous by John Searle, as to whether machines could have 'syntax, but no semantics'. This, as we'll see, is at the heart of his Chinese room argument, and indeed a variant of this view remains popular today. Watching ChatGPT hallucinate, in grammatically perfect English, about non-existent mathematical theorems encourages the thought we're face-to-face with an entity with syntax but not semantics, and so, in addition to vindicating Searle and those who come after him, would give the engineers a task: figure out semantics to make LLMs reliable!

Unfortunately, as I aim to show, this vindication-cum-roadmap is not the way to go. Our best evidence about semantics suggests we can't cleanly draw a syntax-semantics distinction. And our best theories about LLMs, although I won't press the point so much here, suggests the same. The 'syntax, but not semantics' assessment of LLMs doesn't withstand scrutiny.

To anticipate, the idea is simple. If we consider our best theories of semantics, the ones taught to students and featuring in linguistics journals, we don't get clear distinctions. We see again and again that to develop adequate semantic theories requires syntactic commitments; we can't separate them out. Our best theories are hybrid syntactic-semantic theories, and that seems unavoidable. Expecting we can parcel out LLM behaviour into a syntactic and a semantic component is unrealistic.

Not only that. I spoke of best theories, plural, intentionally. There are several live options that have their supporters. But it is notable that they involve notably different views of the syntax-semantics distinction. The distinction accordingly is unclear both intra-theoretically (we can't divide the theories into a semantics and a syntactic part) and inter-theoretically (the theories fundamentally disagree on syntax and semantics). We shouldn't speak about syntax and semantics unless we commit to a framework, but to do so would be rash given there is no consensus as to which is the right ones.

The plan is as follows. In the first section, I present influential contemporary work that can be seen to defend the syntax but not semantics thesis, showing how this line of thought goes back to Searle. Then I consider a recent critique of Searle, published in this journal, by Jaroslav Peregrin. Peregrin doesn't consider contemporary versions of the Searleian view, and so one might worry that his position doesn't speak to the debate today. The bulk of the paper will be devoted to showing that an updated version of his view does, making in detail the argument sketched above.

Searle, forty years, Bender *et al.*

Let's colourfully call the idea that artificial entities have only syntax and not semantics *thesis*. Thesis isn't new. Arguing for it is actually the aim of one of analytic philosophy's most famous thought experiments: Searle's Chinese room, which purports to show precisely that a machine has 'syntax, but not semantics'. The argument for this is that meaning requires intentionality, but because a computer is restricted to "perform[ing] computational operations on formally specified elements" (Searle 1980: 418), "instantiating a computer program is never by itself a sufficient condition of intentionality" (416). The upshot is that we get from Searle a conception of syntax—symbol manipulation—and semantics—intentionality—according to which machines can have the former but not the latter.

It's tempting to think that Searle's Chinese room must be old news. There's been a lot of water under the bridge; contemporary systems are massively different from the good old-fashioned AI of the 80s. And semantics has changed a lot: while intentionality surely still has a role to play in accounting for our understanding of language, that understanding is much more nuanced. One lesson of the externalist tradition, for example, is that intentionality is considerably less important than might be thought.

Surprisingly, though, Searle's shadow looms large over contemporary discussions. Thus for example Meta's Head of Ai, Yann LeCun, in an article co-written with Jacob Browning (LeCun and Browning 2022), consider the possibility that "knowing the right sentences and when to deploy them exhausts knowledge" (which could be viewed as a motivation for the Turing test) before pointing out approvingly that that idea is subject to a 'withering critique', namely … the Chinese room argument! Forty something years later the idea is still being wheeled out as if it were a settled issue.

Not only is Searle referenced by name, but one of the most famous thought experiments in the contemporary literature is a variant of the Chinese room. This comes from an influential article published in 2020, by linguists Emily M Bender and Alexander Koller, who argue that LLMs can only manipulate 'form' and never meaning, where by 'form' they mean:

> any observable realization of language: marks on a page, pixels or bytes in a digital representation of text, or movements of the articulators. We take meaning to be the relation between the form and something external to language (Bender and Koller 2020: 5186-7).

In particular, a meaning, for Bender and Koller, is a pair of a bit of language and a communicative intent to refer to something using that bit of language; there is also 'conventional or standing meaning', that part of meaning that is constant among contexts. To use language one picks an appropriate conventional meaning and the hearer works out what you mean on the basis of it.

In order to make this point they consider a thought experiment, the Octopus story. Two humans (let's call them A and B) are stranded on different desert islands; by good fortune there's a data cable running under the water between the two islands they can use as a telegraph. A smart Octopus is also on the line and listens to their conversations. With enough time, Bender and Koller argue, the

octopus could learn enough about English to be able to pass a version of the Turing test--if the Octopus were to change the wire so one person was removed and the other was talking to the Octopus, the Octopus could convince that person they--the Octopus--were human.

However, this only goes so far. If the Octopus 'learns' when to properly use 'coconut' (for example, learning that 'coconut' often co-occurs with 'eat'), it might be able to answer reasonably when person a says 'what should I eat?' If the octopus says 'coconut', the person might take that as reasonable dietary advice. And on that basis, an onlooker might be impressed with the Octopus's skills.

The problem arises if we consider elaborations of the case. A is faced with an angry bear and asks for advice on how to make a weapon from sticks. The octopus, having only been exposed to unserious chitchat, will have no sense of how 'weapon' works; they won't even know in any sense what bears or weapons, or perhaps even sticks, are. It will soon become apparent that at the other end of the cable there isn't a human.

They conclude:

> Having only form available as training data, it did not learn meaning. The language exchanged by A and B is a projection of their communicative intents through the meaning relation into linguistic forms. Without access to a means of hypothesizing and testing the underlying communicative intents, reconstructing them from the forms alone is hopeless, and O's language use will eventually diverge from the language use of an agent who can ground their language in coherent communicative intents (Bender and Koller 2020: 5189)

The important thing to note is that in many respects this is a Chinese room-type argument. It's about the failure really to bear content of a system that acts, in some respects, in the right way, and the failure, as in the Chinese room, is pinned on the fact that the system lacks a particular mental state (intent or intentionality). It presents a gulf between syntax (form) and semantics (meaning). At least, that's a natural interpretation.

This paper, and perhaps especially its quasi-sequel (Bender et al 2021), are very often discussed in AI circles as one of the main challenges to the claim that contemporary LLMs have semantic capabilities. Indeed, it's noteworthy that even those who dissent seem to agree with the underlying claim. Consider, for example, the work of Deep Mind's Steven Hill and UC Berkeley psychologist Steven Piantadosi, who defend a conceptual role theory of meaning:

> Bender & Koller argue that text-based LLMs will never have meaning because these models lack reference. However, they do not demonstrate that reference is the key to meaning—instead they assume it. …[We've argued] [m]eaning instead seems to come from the way concepts relate to each other. It is these interrelations that LLMs know something about since their internal geometries and trajectories approximate those of humans. Like people

> who don't know that water is H2O and so could not pick it out based on chemical composition, Bender & Koller's octopus lacks some aspects of conceptual role like physical appearance. But, both the octopus and people know other parts of conceptual role that are sophisticated in their own right. If theories about conceptual role are the correct account, then LLMs likely already share the foundation of how our own concepts get their meaning. (Hill and Piantadosi 2021: 5)

Thus they seem to assume that if meaning involves reference to the external world, then LLMs can't refer. That means they accept that LLMs operate on form--they just think that enough form can give one a conceptual sphere in which expressions bear relations to one another and derive their meaning from their place in the sphere.

A Recent Critique

In a recent article in this journal (2021), Jaroslav Peregrin critiques Searle's view. If the above is correct, then contemporary work derives from Searle. Therefore, it's at least possible that critiques of Searle serve as critiques of contemporary work. I will show this possibility is realized: I will extend Peregrin's critique in ways he hasn't done before applying to it people like Bender et al, LeCun, and so on.

Peregrin's argument against Searle is simple: the concepts of syntax and semantics in terms of which he phrases his thesis are equivocal: they can be understood in different ways, and the thesis is more or less plausible depending on the understanding used. Discussing the matter in terms of syntax and semantics gives merely an 'illusion of understanding'.

In order to make this point, he presents us with a couple of alternative linguistic frameworks in which the distinction is understood differently. The Searleian view posits a gulf between syntax and semantics; later views between form and meaning. One way to counterexample such a claim is to show a viable framework in which there is no gulf. And there is such a framework: William Rappaport's syntactic semantics.

The core idea of this theory is that semantic relations are a species of syntactic relations. Just as 'jump' and 'jumped' bear a species of syntactic relation to each other, so, on this view, do 'John', and John. No quotation marks! John, the object, is to be conceived of as something capable of bearing syntactic relations to other words, and the relation of reference becomes a syntactic relation relating two syntactic entities, 'John', and John.

Properly to motivate Rapaport's view is beyond the scope of this article: see e.g. his 2019 for this. Important is that syntactic semantics is at least possibly a contender for a theory of the syntax semantics architecture and one on which one would easily get from form to meaning.

You might think—especially given this extremely compressed presentation—that syntactic semantics is too weird to make us reconceive the syntax/semantics distinction. But Peregrin shows that more mainstream theories lead in the same direction. He defends an inferentialist theory of meaning somewhat like the Piantadosi Hill view mentioned above. For such a view, the meaning of a word like 'and' is exhausted by the twin facts that---for sentential values of the variables---from *a and b* one can infer *a* and from *a*, *b* one can infer *a and b*. Provided our conception of inferring here appeals

only to form or structure (which will be more or less plausible relative to different frameworks in the philosophy of logic), inferentialism will come out as a theory on which one can get from syntactic facts to semantic facts. Inferentialism thus serves as a possibility proof that there isn't necessarily a gap between syntax and semantics, and one variants of which still command respect today.

Semantics 40 years later

Peregrin's point is well made. But we can extend it. I aim to show that not only can we find theories of language that understand syntax and semantics differently, but that such theories are in fact ubiquitous and mainstream. The developments of roughly the last forty years have seen a range of linguistically informed and technically sophisticated semantic theories which differ, to a large extent, on their view of the syntax/semantics distinction. Not only that—not only do the theories differ between each other, but *even when we consider one in isolation* we don't get a clear syntax/semantics distinction. Looking at our best theories, we see syntax and semantics intermeshing, as syntactic assumptions are made purely for semantic ones.

A respectable position for a contemporary formal semanticist is to think that one of these theories is right, but that neither is clearly and proven right. And so it follows that contemporary semanticists should be equivocal as to which is the best syntax/semantics architecture. We can only assess the Searleian thesis if we take a position on the right formal semantic theory; but we can't do the latter, so we can't do the former.

Generative Semantics

In order to make the points of the remainder of the paper, we need to review, as briefly as possible, the standard assumptions of the generative semantic framework. On my reading, there are three theories that comprise the mainstream: generative semantics, dynamic semantics, and variable-free semantics, although it should be noted that variable-free semantics is less influential than the other two. For that reason I will omit discussion; I hope the reader with knowledge can see how analogous points could be made.

The key twin points are that dynamic semantics arose owing to perceived failings of the generative framework, and that these failings pertain to issues at the syntax/semantics interface. We'll begin by presenting generative semantics, following its textbook presentation in Heim and Kratzer 1998. It is based on the following idea:

> **Idea.** The meaning of a complex expression is determined by performing function application (and any other necessary operations) on that expression's syntactic constituents, as those constituents are revealed by syntactic theory.

That, readers aware of generative semantics will see, is a bit rough (I should have talked about branching trees rather than constituents and there are other composition procedures that aren't function application), but enough for us. The twin ideas of syntactic constituency and function application, respectively a syntactic and a semantic idea, are what we need.

The idea of a constituent we can leave on an intuitive level. Consider:

- The president saw Smith

There's a sense in which certain parts of this sentence 'go together'—form natural units—in a way that others don't. 'President saw' isn't a constituent in a way that 'the president' is. One way to test for this is to try conjunction tests: if one can conjoin a given expression with something similar of the same syntactic type, it's likely a constituent. Then note:

- The president and the vice-president saw Smith
- #The president saw and vice president greeted Smith

We use tests like these to work out a parse or phrase-structure tree for the sentence which captures the syntactic relationship between expressions. We might have something like the following, where the exact details don't matter but the visual fact that brackets bracket expressions that seem to form units does matter:

[_s [_np [the president] [[np [and] [the vice-president]] [_vp saw Smith]]

To see the second idea, function application, consider the semantic relationship between 'the' and 'president'. Arguably, we know two things about the meaning of the expression: it stands for an object in the world, and its meaning is dependent on its parts.[1] So here's a challenge for us: give rules saying how the meaning of its parts determine what it means, by assumption an object.

Here's how we do it. Assume, reminding ourselves of predicate logic, that we have three sorts of entities. We have objects, like Obama or Paris; truth values (the True and the False); and functions. (In doing logic we work with sets but we can translate from set-talk to function-talk easily: a set whose members satisfy a condition ϕ, {ϕ(x)}, can be thought of as the function mapping an object to True provided ϕ(x)). Anything that maps an entity to an entity counts as function, *and thus as an entity*. This recursive (albeit informally specified) definition of entity will do a lot of the work for us.

Then consider

- Obama is in Paris

We know a few things: 'Obama' stands for Obama. The sentence stands for a truth value. The whole is dependent on its parts. And since 'is in Paris' isn't either an object or a truth value, it must be a function, and a little reflection reveals it to be that function mapping an object to True provided that object is in Paris.

Let's notate entities of object type as e, of truth value type t, and functions from a type f to a type g as <f,g>. 'Obama' is type e; 'Obama is in Paris' type t, and 'is in Paris' is <e,t>. Note that both the

[1] The reader familiar with 20th century philosophy of language will realize it's controversial that definite descriptions stand for objects. An alternative quantificational analysis is readily available, and will be hinted at below.

range and the domain of functions can themselves be functions, per our definition. These are all acceptable types: <<e,t>,t>; <e,<e,t>>; and so on.

It turns out we can capture a lot of language just with these ideas. Consider, moving beyond the capacity of first order logic

- The president is in Paris

Remember the rules of the game: we assign types to parts to generate the meanings of wholes. Consider just 'the president'. Still assuming it stands for an object, it is type e. And we know that 'is in Paris' is <e,t>. But then we have enough information to work out the meaning of 'the'!

It's not a truth value, and it's not an object. So it must be a function. Moreover, its output must be an object, so we have <?,e>. Finally, we know the meaning of what it applies to—it's <e,t>. So we get <<e,t>,e>. 'The' is something that takes a function and returns an object.

We can extend this. What about 'loves'? Well, think about it—'loves' takes an object (its grammatical object) and another object (its grammatical subject) to give a truth value. So we want something like <e and e, t>. For reasons we don't need to get into, we amend that and say instead it's an <e,<e,t>>. We treat 'loves' as a function that yields a function given an object as opposed to something yielding a truth value given a pair of objects. What about 'everybody'?

Here things get tricky. For concreteness, consider:

- Everyone laughed
- John laughed

If you think about it, hopefully you can see the problem. We know what laughed is <e,t>. It looks for an object and gives a truth value. For our second sentence, that's fine. But surely 'everyone' doesn't stand for an object!

We can solve this problem. But—and this is the crucial idea for us—to do so requires taking positions in syntax. Our desire for a neat semantic theory has syntactic costs, costs that others refuse to pay.

Here's how we do it. We first note that it's plausible, if still a subject of contention among syntacticians, that expressions *move*: appear in places other than they appear at deep structure. Deep structure is the structure revealed after syntactic analysis: for example, it encodes constituency facts that aren't visibly apparent on the surface (see any standard introductory syntax textbook, such as Carnie 2021). We need movement to explain things like wh-movement. If you consider a sentence like:

- Who did he see?

There's a subtle oddness: the verb 'see', which normally has a term denoting an object in its argument place (as in 'He saw John' or 'He saw someone'), doesn't. But if you think about it, we know what should fit that argument place: the word 'who'. After all,

- You saw who?

Is just about an okay English sentence in some contexts. Moreover, this point is strengthened by considering cross-linguistic data. Thus Chinese languages are what linguists call 'wh-in-situ' languages—question word don't move. Here's an example from Mandarin:

- Nǐ kànjiàn shéi?

- You/saw/who

To reconcile these facts we posit that at deep structure, 'who' appears in argument position but it is *moved* for the surface structure. It is moved to the front on the surface but the sentence what's called a 'trace' in its original position. A trace is a bona fide syntactic expression and so can fill the gap to the right of 'saw'. It is linked with the quantifier that moved to indicate that the quantifier has been moved. So we assume that our surface sentence is generated as so:

- You saw who

We move 'who', leaving behind a trace. We also index the question word and the trace, where indexes are to be conceived of as a syntactic thing as well:

- Who$_1$ you saw ~~who~~ t_1

This is what we assume the underlying syntactic structure of the sentence is. Semanticists, noting that movement appears to be needed to capture syntactic facts, put it to work to deal with our problem. First, we assume that 'everybody' moves and leaves a trace

- Everybody$_1$ t_1 ~~everybody~~ laughed

We've solved one problem but replaced it for another. Now there's no problem with 'laughed'—it has an object of the right type, but the whole sentence seems both syntactically and semantically ill formed, as ill formed as 'everybody grass is green'. To solve this new problem---we add more syntax! In particular, we add something called a lambda binder. A lambda binder is an expression that takes a phrase, of type y, containing a trace and returns a function from the type of the trace to y.

So we have

- Everybody$_1$ λ_1 t_1 laughed

"λ_1 t_1 laughed" denotes the function that maps x to True provided x laughed. And with that our analysis is almost over. We can know play 'guess the function' again—the whole is T, the verb phrase is <e,t>, 'everybody' isn't e. It must accordingly be a function that takes <e,t> to t. And so it is. It's the function that maps an <e,t> to true provided everybody is mapped by that function to true.

What's the Point?

Phew! But what does all this have to do with AI, and in particular the syntax/semantics question we're concerned with? The first thing is the thorough entangledness of syntax and semantics. To solve our problems, we posit a lot of hidden structure. Although the textbook is called *Semantics In Generative Grammar*, a better title might be *Semantics (And Some Needed Syntax) In Generative Grammar*.

Less facetiously: say you buy the generativist approach---and I remind the reader that it is absolutely mainstream, and we've been following a textbook presentation---you're implicitly buying a framework in which syntax and semantics are entangled, in the following sense: our theory of syntactic structure is partially determined by facts about semantics, such as facts about how quantifiers work. We do not have a picture of syntax on one side and semantics on the other.

Imagining a complete syntactic theory in the sense of a set of formal rules that make no mention of meaning is to be doing something different from what linguists working with meaning actually do.[2]

Of course, the generativists could be wrong. I would note that there doesn't seem to be a contender in terms of empirical coverage, but it's worth exploring the consequence. If the generativists are wrong, who is right, and what do those right people have to say about syntax and semantics?

Dynamic Semantics

Although it's right to say the generative approach is mainstream, it has a contender, one whose basics and motivation we'll explore here. Recall the crucial idea of compositionality: an expression's meaning is determined by its constituents, where constituency was a question of going together as revealed by things like the conjunction tests.

Dynamic semantics is based on the observation that the meaning determination relation is not so simple, because expressions that don't form a constituent interact in a systematic way to determine the meaning of expressions of which they are part, where expressions here can be considerably 'larger' than countenanced by the generative approach, and indeed can comprise several sentences. This observation leads to a rather different style of semantic theory.

First, let's explain the somewhat forbidding idea of the previous paragraph. It's actually simple. Consider:

- John looked at his son and dog Barker. Proud of his family, he took a photo.

The important fact is that the name 'John' and the pronouns in the second sentence ('his', 'he') go together, in the sense that the claim the sentence makes upon the world has to do with a single object who looked at his son and dog (expressed in the first sentence) and the same object, proud of his family, took a photo. 'John', 'his', and 'he' form a semantic unit that, one might think, we need to account for.

The problem is that those expressions don't from a syntactic unit. No constituent of any sentence in what we call the discourse—the set of sentences—contains those expressions. But our generative theory of meaning determined the meaning of a whole part on the basis of its constituents. It seems our theory of meaning doesn't work for these *long-distance dependencies.*

One of the goals of dynamic semantics is to make a theory that works for them. But in order to do that we need to modify how we understand both semantics and syntax, meaning and form. Intuitively, we need to somehow make the things interpreted by semantics 'bigger'—big enough that whole discourses can be assessed in one go. That is idea one: for dynamic semantics, the primary bearer of truth-like semantic properties is sets of sentences. Idea two is that we need a syntactic

[2] It's important to point out that this is controversial. While I think one can't deny that we need syntactic assumptions to generate a systematic semantics, many are sceptical about such semantics. If one asked a syntactician whether they were happy with the sort of assumptions we've looked at, the answer would probably be no.
I don't think this affects the overall point too much. It might be that the project of semantics adumbrated here is misguided. But this approach surely is the best we have—by far it captures the most data. If it is misguided, then perhaps semantics as an enterprise is misguided, and so 'semantics' is empty or something like that and so our distinction is ill-founded. So we get our conclusion by a slightly more circuitous route.

representation of the discourse that captures the fact that elements far separated from one another need to be semantically interpreted the same way.

There are two ways of doing this: file-change semantics (Heim 1981) or its predecessor dynamic predicate logic (Groendijk and Stokhof 1991), and the discourse representation theory initially introduced by Hans Kamp (1981). At the heart of DRT is the idea of a discourse representation structure. Roughly, this is a representation of the discourse, updated as new sentences are uttered. The basic structure of a DRS is as so:

[x: Fx]

That is to say, it encodes information about discourse referents (indicated with x), and properties of those discourse referents (open sentences of a formalized language with a free variable). Consider

- John laughed and Mary danced. Then he sang.

[x, y.: John(x), laughed(x). Mary(y), danced(y)]

The second sentence leads to its own drs [x. sang(x)] and thereafter we merge the two, yielding:

[x,y. John(x), laughed(x), sang(x). Mary(y), danced(y)]

It is this updating that enables us to account for the semantic interrelatedness of the name and the pronoun. Note how two syntactically distant expressions ‘laughed’ and ‘sang’ come out to be collocated in the DRS associated with the x variable.

Once we have a DRS for the whole discourse, we can evaluate it for truth. Note this is very different from normal semantics where we evaluate sentences in term. We say that a DRS is true provided we can map the variables it contains to entities in the world satisfying the conditions predicated of the variables. In DRT, there’s a default, text-level, existential quantification that we apply to DRSs to get truth.

There are some things to note. Firstly, it seems reasonable to analogise the DRSs to syntactic representations, for the simple reason that a role of syntactic representation is as the input to semantics. Secondly, and just as we saw with the generative framework, it’s nevertheless a semantics-flavoured sort of syntax. Generally, the very existence of DRSs as a sort of representation is owing to the felt need to capture in semantics facts about anaphoric coreference (this is not unquestionable: one could hold that such facts are pragmatic facts). Specifically, the fact that we end up with the DRS above, with ‘laughed’ and ‘sang’ together predicated of the same variable, is based on the semantic judgement that ‘he’ and ‘John’ corefer.

So I think we can draw the same conclusion about dynamic semantics as about generative semantics: the distinction between semantics and syntax can’t be clearly drawn. The DRS of the dynamic semanticist is best viewed neither as a semantic nor as a semantic object, but one determined by both syntax and semanticists, a sort of hybrid.

Conclusion

I think there’s accordingly a good case to be made that the syntax/semantics distinction, as manifest among mainsteam semantic theories, is not a sharp one. Different theories make different syntactic or syntax-like assumptions in order to capture semantic desiderata. Intra-theoretically, syntactic properties can’t be pulled apart from semantic ones: lambda binders, although syntax entities, are posited purely for semantic reasons; DRSs, levels of representation akin to syntactic trees, have the

structure they do to capture anaphoric (semantic) dependency relations. And inter-theoretically, the overall architecture arrived at is markedly different: generative semantics and dynamic semantics have fundamentally different conceptions of how form and meaning relate. We can suggest a tentative conclusion: a hard and fast syntax/semantic distinction is not fundamental to our best semantic theories.

And if that is so, why should it be fundamental to our theories of artificial semantics? It is hard to think of an answer that isn't going to be question begging. One might think but surely computers only operate on form or syntax. But the whole point is that this "only" is unjustified. A putative human operating only on syntax is either operating on logical forms containing lambda binders or DRSs. But these, to repeat, are quasi-semantic entities, entities whose existence is necessitated by semantic facts.

Let me end by noting that the picture offered here is in fact comports very well with our best understanding of the technical aspects of deep learning. Although this is work for a separate paper, recall that two of the key parts of contemporary transformers are word vectorization and attention. Vectorization is the by now familiar fact that we can encode words as large dimensional arrays of numbers, where intuitively the dimension reflects a feature of the world that's relevant to its use, where this feature can either be syntactic or semantic. If one takes word vectorization as a theory of something like the primitive meaning types of a language, then arguably contemporary work too speaks against their being an important syntax/semantics distinction. Secondly, the attention mechanism in transformers is, in the words of Andrej Karpathy, a 'communication mechanism', enabling information from earlier in a sequence to inform the representation of meaning later in the sentence. A translation transformer, for example, might consider the sequence 'she jumped' and when encoding 'jumped' seek to encode in the representation of 'jumped' information from earlier in the sentence like that 'she' occurs before it. So it might represent that occurrence—roughly—as something like she+jumped, so that when translated to a language that marks gender on verbs, it would pick the right translation. But if that's so, the attention looks like an operation rather similar to the sort we see in dynamic semantics, capturing as it does semantic dependencies across syntactic distance.

This, accordingly, is our final conclusion. It appears the best work in linguistics doesn't have room for a syntax/semantics boundary. And it appears the best work in deep learning doesn't, either. And so I think we have very strong reason to jettison that distinction from our theorizing, and thus to reject the line of criticism from Searle to Bender and Koller that relies on it.

Bibliography

Bender, E. M., & Koller, A. (2020, July). Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 5185-5198).

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021, March). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜 . In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency* (pp. 610-623).

Groenendijk, Jeroen & Stokhof, Martin (1991). Dynamic predicate logic. *Linguistics and Philosophy* 14 (1):39-100.

Heim, Irene (1982). *The Semantics of Definite and Indefinite Noun Phrases*. Dissertation, Umass Amherst

Heim, Irene & Kratzer, Angelika (1998). *Semantics in Generative Grammar*. Blackwell.

Kamp, Hans (1981). A Theory of Truth and Semantic Representation. In P. Portner & B. H. Partee (eds.), *Formal Semantics - the Essential Readings*. Blackwell. pp. 189--222.

Peregrin, Jaroslav (2021). Do Computers "Have Syntax, But No Semantics"? Minds and Machines 31 (2):305-321.Searle, John (1980). Minds, brains, and programs. Behavioral and Brain Sciences 3 (3):417-57.

Piantadosia, S. T. Modern language models refute Chomsky's approach to language. Preprint, https://lingbuzz.net/lingbuzz/007180/v2.pdf

Rapaport, W. J. (2019). Computers are syntax all the way down: Reply to Bozşahin. *Minds and Machines, 29*, 227–237.