ChatGPT

The Chinese Room, Modern AI, and the Syntax–Semantics Divide: Philosophical and Historical Perspectives

Searle's Chinese Room Argument

Background and Motivation: John Searle introduced the Chinese Room argument in 1980 to challenge claims of "strong AI" – the idea that a suitably programmed computer **literally** has a mind and understands language, not just simulates understanding [1]. At the time, AI researchers like Schank and Abelson had developed programs that answered questions about stories, and some claimed these programs **"understood"** the stories. Searle, a philosopher of mind, was skeptical. Drawing on the distinction between **syntax and semantics**, he formulated a thought experiment to show that manipulating symbols (syntax) is not enough for real understanding (semantics) [2] [3]. His broader aim was to refute the *Computational Theory of Mind* (a form of functionalism that equates mind with information-processing) by illustrating that **formal computation alone** cannot yield conscious understanding [4] [5].

The Thought Experiment: Searle asks us to imagine a person (Searle himself, an English speaker) locked in a room with a set of rules for manipulating Chinese characters. People outside the room pass in questions in Chinese; the person inside uses the rulebook (written in English) to correlate the Chinese symbols and produce appropriate Chinese answers, which he then passes out. To the external observers, the answers are indistinguishable from those of a native Chinese speaker [3] [6]. Crucially, *the person in the room knows no Chinese* – he is simply following formal rules, or a "program," for symbol manipulation. From the outside, it **appears** the room understands Chinese, but as Searle notes, "I do not understand a word of the Chinese stories... I still understand nothing" of the content [7]. The thought experiment thus separates *behavioral performance* from *actual understanding*. No matter how fluent the responses, **all the person (or computer) is doing is manipulating uninterpreted symbols** based on syntax, with no grasp of their meaning [8] [9]. Therefore – and this is Searle's key point – running a program that produces intelligent-looking outputs **is not the same as having real understanding or mental states**. The computer, like the man in the room, "has nothing more than I have" in the scenario, which is to say, only formal symbol processing without any semantics or intentionality behind it [10] [11].

Core Reasoning (Syntax vs. Semantics): The Chinese Room argument rests on two key claims: (1) *Brains cause minds*, and (2) *Syntax doesn't suffice for semantics* [2]. Searle later summarized the logical structure of the argument as a set of axioms and a conclusion [12]:

1. **Programs are formal (syntactic).**
2. **Minds have mental contents (semantics).**
3. **Syntax by itself is neither constitutive of nor sufficient for semantics.**
   Therefore, no purely syntactic program by itself can be *mind* or can produce understanding – **programs are not sufficient for minds** [12].

In other words, computers execute formal symbol manipulations (syntax), whereas minds attach **meaning** to symbols (semantics). Searle argues you cannot get semantics *just* from syntax; no amount of shuffling symbols according to rules will magically produce understanding, in the absence of a mind that **knows what the symbols stand for** [13] [11]. This undercuts the strong AI claim that implementing the right program is enough for genuine understanding or consciousness. Searle doesn't deny that machines *can* think in principle (after all, brains are biological machines that produce minds [14]). His point is that merely simulating mental processes with a program is not the same as *duplicating* a mind. He famously quips that if he doesn't understand Chinese by executing the steps of a program, neither does any digital computer running a program, **"because no computer, qua computer, has anything the man does not have"** – namely, intrinsic understanding or intentionality [15].

Replies and Counterarguments: Searle's thought experiment provoked numerous responses from AI researchers and philosophers, many trying to rescue the intuition that a computer could *in principle* understand. Searle initially discussed several standard replies and offered rejoinders to each [16] [17]:

- **The Systems Reply:** Even if the person in the room doesn't understand Chinese, perhaps the *whole system* (the man + rulebook + papers) does. Understanding, on this view, is an emergent property of the entire system executing the program. Searle's rebuttal: let the person internalize **all** parts of the system – memorize the rules and do all the steps in his head. He would still have no understanding of Chinese; therefore, the system as a whole has nothing more than the person has [18] [19]. As Searle puts it, it's *"just ridiculous"* to say that *he* doesn't understand but the *conjunction* of him and some paper might [20]. If the individual components lack understanding, combining them doesn't magically create a mind. (Searle also warns this reply would imply "mind is everywhere" – even our stomachs could be said to understand digestion if we described their input-output behavior as information processing [20].)

- **The Robot Reply:** This response grants that a disembodied computer (or the room) lacks understanding because it has no sensory connection to the world. It suggests giving the AI a body: put the computer in a robot that perceives and interacts with the environment, grounding the symbols in real-world referents. Then, perhaps, the AI **would** understand (by associating symbols with objects/events via perception) [21] [22]. Searle's rebuttal: we can imagine the entire Chinese Room *inside* a robot. Suppose cameras feed visual Chinese inputs into the room and the outputs of the room control the robot's limbs [23]. From the outside, the robot now behaves as if it "understands" (it can talk about what it sees, etc.). But inside, Searle is still just shuffling symbols by rule; he *gains no understanding* of what the symbols mean [24]. Merely coupling the system to the world via sensors and motors adds causal links, but as long as the internal process is purely syntactic, Searle insists, *no genuine semantics emerges* [25]. (He also points out that the Robot Reply actually **concedes** Searle's point in a way: it implies that symbol manipulation alone is insufficient, and **something extra** – sensorimotor grounding – is needed for understanding [26].)

- **The Brain Simulator Reply:** What if, instead of manipulating abstract symbols about stories, the program exactly simulated the firing of neurons in a Chinese speaker's brain? If the computer's internal operations mirrored an actual brain's activity when understanding Chinese, wouldn't that produce understanding? Proponents argue that such a fine-grained simulation **would** have to understand (or else we'd deny a native Chinese speaker understands) [27]. Searle's rebuttal: even a neuron-level simulation is just **formal** symbol manipulation of a different kind. He illustrates this with another variant: imagine the man in the room now operates a vast set of water pipes and valves

arranged like a brain's circuitry; when given Chinese input, he follows the program to open and close valves, resulting in a Chinese output [28]. This elaborate plumbing still has *only syntax*: it simulates the formal structure of neuron firings but nowhere is there a mind that **knows** Chinese. *"The man certainly doesn't understand Chinese, and neither do the water pipes,"* Searle observes [29]. Thus, even perfect simulation of brain processes (without a brain's actual causal properties) fails to produce semantic understanding – it's only *syntactic* duplication of the pattern, not a mind.

- **The Combination Reply:** Perhaps if we combine *all* the above – a brain-level simulation running inside a robot with sensory-motor capabilities, etc. – surely then the system would understand? Searle's reply: *zero times zero is still zero*. Piling on more simulations and connections (vision, motor, neural detail) doesn't create understanding if each component individually has no understanding [30] [31]. At best we might be *convinced* the robot has a mind if we didn't know the internal details, but if we do know it's all formal programs, we have no reason to ascribe it genuine intentionality [32].

- **The Other Minds Reply:** We normally judge other people understand language by their behavior. So if a computer behaves indistinguishably from a fluent speaker (passes a Turing Test in Chinese), shouldn't we attribute understanding to it, by parity of reasoning? Searle's reply: The argument isn't about **how** we know others have minds; it's about **what** we are attributing when we say someone understands. We attribute mental states because we assume there is more than just brute behavior internally. The Chinese Room thought experiment shows that *the same behavior could arise from mere symbol juggling without any mental state* [33] [34]. So *if* we knew a system was just executing a program, we would not be justified in saying it understands, even if its answers are correct.

- **The "Many Mansions" Reply:** Some concede Searle's point that current programs don't have understanding, but argue that in the future **other methods** (beyond formal symbol processing) might give a computer a mind – effectively redefining "strong AI." Searle calls this changing the subject: if AI is no longer about running formal programs, then his objection doesn't apply because *strong AI was originally defined* by the claim that mental processes **are** just computational processes on formal symbols [35] [36]. Redefining strong AI to "whatever method produces cognition" trivializes the original thesis (and dodges the challenge rather than answering it).

Searle stood firm that none of these replies rescue the idea that *mere* computation suffices for understanding. The intuitive appeal of these replies (especially the Systems and Robot replies) ensured the Chinese Room became one of the most debated thought experiments in cognitive science and philosophy of mind [37]. By the early 1990s, over a hundred articles had been published on it, and computer scientist Pat Hayes joked that *"Cognitive Science [was] the ongoing research project of refuting Searle's argument."* [38]. The argument's longevity is due in part to its **simple clarity** and the fundamental issues it raises: What counts as understanding? Is the human mind just a symbol processor or is there an extra ingredient (biological, causal, or otherwise) needed for meaning? As Steven Pinker noted, the Chinese Room became so notorious that it pervaded discussions well beyond academia [38].

Support and Criticisms: Searle's stance has both supporters and detractors across disciplines. Supporters often invoke the **intentionality** gap – the idea that computational processes have only *derived* meaning (we interpret the symbols), whereas human thoughts have *original* intentionality (intrinsic aboutness). Searle's later work connected the Chinese Room to the idea that **consciousness** (or at least *causal biological processes* in brains) is required to produce genuine understanding [4] [14]. He maintained that while machines *could* think, the machine would need causal powers equivalent to brains – simply implementing a

program won't do it [14]. This view (sometimes called *biological naturalism*) holds that **brains cause minds** in a way that formal computations cannot, because brain processes embody more than syntax (they have biochemical, causal properties that give rise to semantics) [13] [39].

Critics of Searle, especially AI researchers, often reject the intuition that the person in the room captures all that's happening. They argue that Searle's experiment begs the question by intuitively denying the system any understanding, or that it sets up an unrealistic scenario (no real AI is isolated with just symbol rules in that manner). The **Systems Reply** in particular has been defended by many: they contend that *if* a system as a whole behaves intelligently, we may as well say it **understands**, even if the subsystem (the man) doesn't. This is consistent with a functionalist view: what matters is the organization of the system, not the material or the single component. Searle counters that this is just attributing a "mind" to the *software* level without justification – unless there's something in the system qualitatively different from the man's rule following, we haven't explained understanding. Some philosophers (e.g. Dennett and Hofstadter) suggested that Searle's intuition pumps (like the room) mislead us, and that if a program passes a robust Turing Test, denying it has any mental state might be a form of "bio-chauvinism." They argue that what Searle's scenario really shows is the limitations of **introspection** – of course the man doesn't feel he understands Chinese, but the **man is just a part** of a larger system that *does* generate understanding at the level of the whole (analogous, perhaps, to how a single neuron doesn't understand the brain's thoughts) [40] [41].

Historical Context: It's worth noting that Searle's argument has intellectual precursors. In Leibniz's *Monadology* (1714), there is a thought experiment known as **Leibniz's Mill**: if we could build a machine (like a giant mill) that thinks, and we walked inside it, we would find only mechanical parts pushing each other, "and never anything by which to explain a perception." Leibniz concludes that you won't find consciousness in mere mechanism; the explanation for perception must lie in the simple substances (monads), not the complex machine [42] [43]. Searle's Chinese Room mirrors this idea: no matter what computations are occurring, if you "zoom in" on the physical processes (be it moving gears or shuffling symbols), you won't see understanding there – mechanism alone doesn't account for mind [43]. Another antecedent is Turing's own insight about a "paper machine" (from his 1948 essay *Intelligent Machinery*). Turing imagined a person simulating a chess algorithm with pencil and paper, following rules without any real grasp of chess – effectively an analog of the Chinese Room for chess playing [44] [45]. This highlights questions of **agency** and understanding: is it the human "computer" who is playing chess, or just implementing the program? Turing used this to argue that what matters is the program, not the implementer's understanding, whereas Searle uses a similar setup to argue that **without understanding in the implementer (or elsewhere), claiming the system 'knows chess' is misguided** [45] [46].

Moreover, Searle's critique of AI had allies in earlier thinkers like **Hubert Dreyfus**, who in the 1960s and 70s argued that human intelligence involves intuitive, embodied skills that cannot be captured by formal rules. In *What Computers Can't Do* (1972), Dreyfus identified "problematic assumptions" in AI – notably the belief that the mind is like a digital computer and that all understanding can be made explicit in symbolic rules [47] [48]. Dreyfus (influenced by phenomenology) pointed out the importance of context, background commonsense, and the body in human cognition – factors that a rule-bound AI lacked. Searle's argument hits a related note: that **meaning** arises from *more than* formal symbol manipulation, though Searle locates the extra ingredient in the biological brain's causal powers, whereas Dreyfus located it in embodied being-in-the-world [49].

After decades of debate, the Chinese Room argument remains a staple of introductory philosophy of mind and AI courses, as a stark reminder of the **syntax–semantics gap**. It poses a challenge to any claim that

passing a behavioral test (like the Turing Test) is equivalent to *actual* understanding. Even today, as AI models produce ever more fluent language, Searle's thought experiment is frequently invoked: we have chatbots that can converse like a human, but are they **truly understanding** or just extremely sophisticated symbol manipulators? This segues directly into modern discussions of AI, as we examine how contemporary models deal with syntax and semantics, and whether Searle's skeptical insight still applies.

Modern AI Models and the Syntax–Semantics Divide

Modern AI language models – especially **large language models (LLMs)** like GPT-3/GPT-4, BERT, or Google's PaLM – have achieved astonishing proficiency in generating and interpreting language. These neural network models are trained on enormous corpora of text and can produce outputs that read as coherent, contextually relevant, and often even insightful. Under the hood, however, their operation is *statistical* and *syntactic* in nature: as David Chalmers describes, such models are **"huge artificial neural networks trained on text to predict what the next word in a sequence is likely to be."** GPT-3, for example, has 175 billion parameters and was trained on virtually all of the Internet's text; it's essentially a massively scaled-up predictor of plausible word sequences [50]. During training, the model adjusts its internal weights to minimize prediction error, picking up on patterns, correlations, and structures in language.

Syntax in LLMs: Despite not being explicitly programmed with grammar rules, LLMs *learn* syntax implicitly. They develop an internal representation of language structure through exposure to countless examples. For instance, they learn that in English, adjectives usually precede nouns, that verbs must agree with subjects in number, etc., simply by statistical reinforcement. The transformer architecture (used by GPT, BERT, PaLM, etc.) uses an attention mechanism that can capture long-range dependencies in sentences, allowing the model to handle complex syntactic constructions (like nested clauses or subject–verb agreement across intervening phrases). In practice, models like GPT-3 can generate grammatically well-formed sentences and even *transform* sentences (e.g. turning active voice to passive), indicating they have abstracted some syntactic rules. They have also demonstrated the ability to parse or at least **interpret** the structure of input: for example, instructing GPT-4 to add commas or to explain the parts of a sentence often yields correct identifications of nouns, verbs, clauses, and so on – a sign that syntactic relationships are embedded in its learned representations. In short, contemporary AI models excel at **formal linguistic competence**: they can mimic the patterns of language with very few grammatical errors, suggesting that the networks encode a lot of syntactic information (word order, phrase structure, punctuation rules, etc.) as a byproduct of their training.

Semantics in LLMs: The more contentious question is whether these models have *semantic understanding* of the language they process, or whether they are simply manipulating symbols (words) based on distributional patterns (analogous to Searle's room). On one hand, modern AI models do capture aspects of semantics in their vector representations. For example, in a model like BERT or Word2Vec (an older word embedding model), words with similar meanings end up with **similar vector embeddings** – the model "notices" that *king* is used in contexts similar to *queen*, or that *Paris* often co-occurs with *France*, etc. This is a reflection of the **distributional hypothesis** (pioneered by linguist J.R. Firth): that words used in similar contexts tend to have related meanings. Indeed, *distributional semantic* models have been a major approach in computational linguistics – and neural LLMs are essentially an advanced form of that, building high-dimensional representations that encode semantic relationships gleaned from usage. These models can even perform analogy-like reasoning in embedding space (the famous *king – man + woman ≈ queen* result from word embeddings), and they often possess a vast store of world knowledge (since they've read billions

of sentences). For example, GPT-4 "knows" that the capital of Japan is Tokyo, or that sushi is a type of food, because those facts are buried in the statistical patterns of text it consumed. In that sense, one could argue the model **represents** semantic knowledge – it has associations that correspond to real-world concepts and facts.

However, critics argue that what LLMs have is *at best* a simulacrum of semantics, not genuine understanding. The model **does not** *know* in any human-like way what Tokyo or sushi actually are; it only knows how these words relate to others in sentences. As one AI ethics researcher put it, *"GPT-3 is not a mind, and not entirely a machine. It's something else: a statistically abstracted representation of the contents of millions of minds, as expressed in their writing… When GPT-3 speaks, it is only us speaking, a refracted parsing of the likeliest semantic paths trodden by human expression."* [51] . In other words, an LLM is like a giant collage of the text it has seen, regurgitating and remixing human-generated content. The *meanings* it seems to express are *ultimately derivative* of human semantic labor. This view aligns with Searle's original intuition: the AI's apparent knowledge is *second-hand*. It manipulates symbols in sophisticated ways but **does it actually *attach* meanings to them?** Or is it just drawing from statistical correlations learned from humans who did have those meanings in mind?

A growing number of researchers and philosophers lean toward the latter interpretation: these models *lack true semantics*. Emily Bender and Alexander Koller (2020) have influentially argued that a system trained **only on linguistic form** "has *a priori* no way to learn meaning" [52] . They emphasize the distinction between **form and meaning**: contemporary language models excel at form (syntax, word frequencies, patterns), but they are not grounded in the world and thus have *no direct grasp* of what the symbols refer to. Bender and Koller caution that we shouldn't be misled by fluent outputs: just because a model can use words in a convincing way doesn't mean it understands the concepts – it might just be very adept at **"matching patterns of words"** [52] . Indeed, these models often **"hallucinate"** – producing false statements that read confidently – which is a symptom of their form-based learning. For example, an LLM might state a wrong historical date or a nonsensical fact with perfect grammar, because it's only drawing from probable word sequences, not verifying against an internal model of reality. This reveals that the model doesn't *know* in the human sense; it doesn't have a built-in semantic model of the world to cross-check truth, it only has the statistics of language.

Philosophers of mind compare LLMs to Searle's Chinese Room: the model has learned to correlate inputs and outputs (questions and answers, prompts and completions) in ways that fool us into thinking it understands. But internally, say the skeptics, it's still **symbol processing without comprehension**. To use Searle's terms, the network's states are **syntactic** (distributed vectors and matrices updating mathematically), and any "meanings" are *assigned by us* when we interpret the outputs. The model itself, lacking perception or intentionality, is argued to be *mindless*. As one commentator quipped, large language models are "stochastic parrots," randomly echoing the data they've seen without any understanding [53] . Proponents of this "stochastic parrot" view conclude that **LLMs are incapable of *actually understanding* language** – they generate plausible sentences by recombining what they've been fed, without any *sense* of what those sentences mean or reference [54] [53] .

A recent academic paper by L.M. Titus (2024) puts it in rigorous terms: LLMs (and similar **statistical occurrence models**) succeed by exploiting correlations between word occurrences and human-encoded semantic relationships, *"not by functioning sensitively to semantic relationships in a way that could support claims to genuine understanding."* These models, Titus argues, **"do not themselves function to represent or produce meaningful text; they just reflect the semantic information that exists in the aggregate"**

of their training data [55] [56]. In other words, any semblance of meaning in their output is parasitic on the patterns of meaning in the human text they absorbed, rather than arising from the AI's own cognitive grasp of the world. Titus calls this the **"Statistical Hypothesis"**: the impressive, meaning-like behavior of LLMs piggybacks on human language usage, but the AI doesn't *actually* understand (it's picking up on statistical cues of meaning without *experiencing* meaning) [57] [58]. The opposing view (which Titus argues against) would be the **"Semantic Hypothesis"**: that these models *do* have some form of genuine semantic understanding. Advocates of the Semantic Hypothesis might point out that the boundaries between syntax and semantics in such models are blurry – the model's internal representations *do* encode semantic distinctions (for example, it *knows* that "eat" is something only living things do to food, so it won't say "the rock ate a sandwich" unless joking). Some AI researchers believe that with more data, larger models, or new architectures, the pendulum might swing toward true understanding. Indeed, GPT-4 has demonstrated a degree of commonsense reasoning and contextual coherence that is startling; it often picks up implicit meaning, humor, and can perform analogies or rudimentary reasoning. Does this indicate a shallow *semantic* competence emerging from all that syntax? Or is it still just a very deep statistical mimicry?

The debate is active in AI ethics and philosophy. On one side, you have those warning not to anthropomorphize these systems. They note how easily humans are fooled: e.g., a Google engineer famously claimed the chatbot LaMDA was sentient because it produced emotionally rich responses, but most experts agreed this was an illusion – LaMDA had no inner feelings or understanding [59] [60]. As Shannon Vallor puts it, *"understanding is beyond GPT-3's reach because understanding is not an act but a lifelong social labor… a sustained project of sense-making anchored in our world"*, something an AI predicting text is not doing [61]. She and others emphasize that real understanding involves being situated in a context, having experiences, goals, and the ability to **"weld new connections"** of meaning through ongoing engagement [62]. LLMs, by contrast, generate each response *afresh* from their training stats, with no continuity of experience or genuine goal – "the essay is not a project or a labor for GPT-3… it's not trying to *say* anything, it's just calculating what a plausible answer looks like" (to paraphrase Vallor's point) [63] [64]. This view resonates with the Chinese Room: the model isn't *trying to communicate* or *thinking about* the question – it has no *intentions* – it's simply producing the next token that best fits its learned distribution.

On the other side, some researchers and philosophers argue that *maybe* these models do have **proto-semantics**. After all, human children learn a lot of language just by statistical observation too, yet we consider them to understand once they can use words meaningfully. If an LLM can carry out a conversation on virtually any topic, answer questions correctly, translate languages, and even solve word problems, at what point do we say it *understands* at least **in some sense**? David Chalmers, for example, muses that GPT-3 forces us to "fragment and re-engineer our concepts of understanding" – perhaps there are degrees or kinds of understanding [65]. He asks pointedly: *"Can a disembodied, purely verbal system truly be said to understand? Can it really understand happiness and anger just by making statistical connections? Or is it just making connections among symbols that it does not understand?"* [66]. These questions he leaves open, but suggests that as AI improves, we might have to refine what we *mean* by "understand." It's possible that an AI could develop an **internal model** of the world purely from text (some argue GPT-4's high-level performance indicates an internal representation of concepts and their relations – a kind of *knowledge graph* implicit in the weights). If so, one could say the AI has *"learned"* semantics – just not in the embodied way humans do. Additionally, efforts like incorporating **multimodal data** (images, videos, audio) aim to give models a form of grounding. For instance, if a model like GPT-4 is fed images along with text, it might connect the word "cat" with actual cat pictures, bringing it a step closer to grounded meaning. Yet, as Titus notes, even multi-modal models can be seen as just extending the statistical trick – they correlate text with pixels, but do they *know* a cat the way a child or even a real cat knows what a cat is? Titus extends his

argument to systems like DALL-E (which generate images from text), asserting that the fundamental issue remains: the AI is leveraging human-created links between images and captions, not truly inventing concepts anew [67].

In summary, **contemporary AI models handle syntax brilliantly and semantics superficially**. They demonstrate that with enough data and parameters, a machine can **approximate understanding** to a degree that fools humans. Yet, according to many philosophers, they are still trapped in Searle's paradigm: they **simulate** understanding rather than possess it. The consensus in AI ethics circles is cautious: we should not treat AI outputs as if they come from an understanding mind, because that can lead to misuse (e.g., trusting AI with decisions or information that it doesn't actually *understand*). As one philosopher quipped, *"GPT-3 is not a mind, but also not just a machine – it's a distillation of millions of human minds. Its prose spurts from an inductive funnel of human language. When GPT-3 speaks, it is only us speaking."* [51]. In effect, current AI is a mirror to human semantic content, not a new independent semantic agent.

That said, the frontier is moving fast. Some researchers are exploring ways to imbue AI with more **grounded semantics** – for example, by integrating robotics (giving the AI a body or simulated environment to experience), or by explicit **symbol grounding** techniques (tying neural network activations to real-world sensors). We are, in a sense, revisiting the old **Robot Reply** to Searle: maybe embodiment plus computation will yield understanding. Indeed, the field of **embodied AI** and multimodal learning is partly motivated by the belief that meaning arises through interaction, not static text. So far, no AI has **fully bridged** the gap, and the Chinese Room argument continues to be a cautionary tale: just because an AI can answer, *"Yes, I understand,"* we must ask *on what basis* is it producing that answer – genuine comprehension, or just pattern-matched chatter?

In conclusion, modern AI forces us to reflect on the syntax–semantics divide anew. It's clear these models *operate* by exploiting syntactic regularities in language. Whether something like semantics "emerges" from this is controversial. As of 2025, the safe claim is that AI **does not yet *understand*** in the rich way humans do – it has no consciousness or intentionality behind its words [14]. But it can **fake understanding** extremely well by leveraging the semantic work done by humans in text. This realization has practical implications (e.g., AI systems can mislead users into over-attributing understanding or agency to them) and philosophical ones (what would *count* as true understanding in an AI, and how would we know?). The conversation between AI engineering and philosophy of mind is very much alive, hearkening back to ideas from Frege to Turing to Searle as we try to chart a course toward machines that might genuinely grasp meaning.

Historical Overview of Syntax and Semantics in Philosophy and AI

The concepts of "syntax" and "semantics" – roughly, *form* and *meaning* – have been central to both philosophy of language and the development of artificial intelligence. Over time, our understanding of these concepts has evolved through contributions from logicians, linguists, computer scientists, and cognitive scientists. Here we'll trace a historical arc: from the foundational philosophical distinctions (late 19th and early 20th century), through mid-20th century developments (like Turing's ideas and Chomsky's linguistics), to the eras of symbolic AI, connectionism, and today's deep learning. Throughout, we'll see a tension between approaches that emphasize **formal structure (syntax)** and those that emphasize **meaning (semantics)** – and attempts to reconcile the two.

Early Philosophical Foundations: Frege, Russell, and Wittgenstein

In late 19th-century philosophy, **Gottlob Frege** pioneered the formal analysis of language, effectively *inventing* modern semantics for logic. Frege drew a now-famous distinction between **sense (Sinn)** and **reference (Bedeutung)** of linguistic expressions. The *reference* of a term is the actual object or truth-value it corresponds to, while the *sense* is the mode of presentation or the cognitive content of the term. For example, "the Morning Star" and "the Evening Star" both refer to the planet Venus, but they have different senses (how they present the object) – which is why "The Morning Star is the Evening Star" is an informative statement, not trivially true. Frege showed that **sense determines reference**: if you know the sense of an expression, you can figure out what it refers to (if anything) [68] [69]. This was crucial for understanding how language conveys meaning. He also introduced the idea that sentences express propositions, and the *meaning* of a sentence can be understood in terms of its truth-conditions – basically, under what conditions the sentence would be true or false [70] [71]. In formal logic, Frege defined a system where the syntax (logical symbols and well-formed formulas) is paired with a clear semantics (interpretations that assign objects to names and truth-values to sentences) [70] [72]. By doing so, Frege **redrew the map for philosophy**: he established that there is a rigorous way to talk about meaning separate from psychological or purely grammatical considerations [73] [74]. Analytic philosophers embraced this, treating the logical *semantics* of statements (especially in mathematics and science) as a primary object of study, distinct from mere verbal behavior.

Bertrand Russell, in his 1905 paper *On Denoting*, built on Frege but introduced his own twist. Russell tackled descriptions like "the present King of France," which posed a puzzle: there is no present King of France (in 1905), so what does the sentence "The present King of France is bald" mean? Frege might say the reference is empty, but Russell offered a **theory of descriptions** that analyzed such sentences into logical form – effectively showing that a seemingly simple subject-predicate sentence actually expresses a complex proposition (involving existence and uniqueness conditions) [75] [76]. In doing so, Russell delineated how **surface syntax** can be misleading and how one must uncover the true **logical form** to get at the semantics. His work also emphasized *direct reference*: while Frege thought every meaningful term has a sense, Russell suggested that some terms (like proper names, if understood as directly referring) connect language to the world without a descriptive sense. This foreshadowed later debates about whether meaning is something in the head (sense) or out in the world (reference).

Russell and Frege together set the stage for thinking of **syntax and semantics as separate components** of language: syntax is the formal structure (which Russell helped unveil with logical form), and semantics is the assignment of meanings to that structure (which Frege's system facilitated). In early analytic philosophy, the dominant view became that a key part of understanding language is giving a **truth-conditional semantics** – specifying how each sentence's truth value depends on the world. Philosophers like Carnap and Tarski later formalized this further (Carnap spoke of **intension** vs **extension**, roughly sense vs reference, and Tarski gave a formal definition of truth in language). All of this was foundational for AI as well, since AI's symbolic systems in the mid-20th century often drew from formal logic.

Ludwig Wittgenstein provides a dramatic twist in this story. Early Wittgenstein (in the *Tractatus Logico-Philosophicus*, 1921) was very much in line with Frege and Russell. He believed that a logically perfect language could **"picture" facts about the world** – each meaningful sentence, in his view, has a logical form that mirrors reality's structure [77] [78]. Any sentence that *has sense*, he claimed, could be translated into logical form; if it couldn't, it was essentially nonsense. This is a strong alignment of syntax and semantics: the syntactic structure of a proposition (in a purified logical language) would reveal its semantic content as

a picture of a possible state of affairs [77] [78]. Wittgenstein's early view thus echoed the idea that language meaning is tightly bound to *formal structure* mapping onto the world.

Later, however, Wittgenstein underwent a famous change of perspective (as seen in *Philosophical Investigations*, published 1953). He came to reject the notion of a strict formal logical language as the key to meaning. Instead, he emphasized the **use of language in practice** – encapsulated in the slogan, *"meaning is use."* On this later view, words and sentences mean what they do *because of how they are used by people in various "language-games"* within forms of life [79] [80]. Wittgenstein highlighted that language is woven into activities and forms of life, and understanding a word is more about knowing how it functions in human practices than about linking it to a precise definition or referent. He pointed out that not all words have a clear reference (e.g. words like "game" or "number" have *family resemblances* across instances rather than a single essence). Also, he argued that **syntax alone is insufficient** to explain meaning: *"One cannot look at the representational dimension of language alone and expect to understand what meaning is."* [80]. In the *Investigations*, he explores how following a rule (syntax) doesn't guarantee meaningful use; meaning arises from the **public, shared practices** that give context to our rule-following [79] [81]. For example, a sequence of letters only counts as a meaningful sentence if there is a convention and practice that invests those letters with meaning. This was a shift to a more **pragmatic** view of semantics: the focus is on how people *actually use* language to achieve understanding.

Wittgenstein's later insights influenced the philosophy of language by introducing the importance of **context, intention, and convention**. In terms of syntax vs semantics: he showed that the same syntactic sentence can mean different things in different contexts (irony, jokes, questions vs statements), and that *understanding* a sentence is not merely a formal exercise but involves grasping the speaker's intention and the language-game being played. This had echoes in AI much later, for example in pragmatic natural language understanding and in critiques of purely formal AI systems (Dreyfus, mentioned earlier, was inspired by similar ideas of contextual, non-formal knowledge).

Turing and Early Computation: Formalism Meets Intelligence

Turning to computation, **Alan Turing** in the mid-20th century approached intelligence from a behaviorist and syntactic angle. In his seminal 1950 paper *"Computing Machinery and Intelligence,"* he proposed what we now call the **Turing Test**: if a computer can converse via text in such a way that a human judge cannot distinguish it from a real human, then *"we should grant that it is intelligent."* [82]. Turing was essentially sidestepping the philosophical definition of *understanding* or *mind*, and instead giving an **operational criterion** based on linguistic performance. This had a profound effect on AI: it suggested that **syntax (correctly generating responses) might be sufficient to ascribe intelligence**, at least pragmatically [82] [83]. Searle later explicitly contrasts his argument with the Turing Test: *passing the behavioral test is not enough*, says Searle, because you could pass it (as the Chinese Room person does) *without* understanding [84] [85]. Indeed, Searle's scenario is basically "what if a computer passed the Turing Test in Chinese? It still wouldn't truly understand." This shows the tension: Turing's stance was more optimistic that *syntax could mimic semantics to the point of indistinguishability*, whereas Searle (and others) worried that semantics involves an inner qualitative aspect not captured by external behavior.

Nonetheless, Turing's work laid the groundwork for **symbolic AI**. He viewed computation as manipulation of formal symbols (Turing machines, after all, manipulate 0s and 1s on a tape). His notion of a "paper machine" – a human computing algorithm steps on paper – underscores that the *mechanism* of computation is syntactic transformation of symbols based on rules [44] [45]. If intelligence could be achieved

by a sufficiently complex program, it would mean human thought is, at root, symbol manipulation. This became the working hypothesis for early AI researchers: they tried to create programs that, via formal rules, would exhibit intelligent behavior (solve problems, prove theorems, understand language). And early successes like the Logic Theorist (1956) or the General Problem Solver (1960) by Newell and Simon, or later expert systems, were all about encoding **symbolic rules** and facts (semantics) into a computer and letting it syntactically derive conclusions.

Interestingly, **early AI** did concern itself with semantics in the sense of knowledge representation. Researchers realized that to solve complex tasks, a program needed knowledge about the world (semantics), not just syntactic rules. This led to the development of *semantic networks* (graph structures of concepts), *frames* (structured representations of situations), and knowledge bases with logical assertions. For example, an early natural language understanding system, **SHRDLU** (Terry Winograd, 1971), could move virtual blocks and answer questions about them. It had a well-defined *micro-world* and a set of semantic definitions for words (like "cube," "table," "bigger than"). SHRDLU combined **syntactic parsing** (it used a grammar to parse commands) with a **semantic analysis** that mapped the parsed sentence to actions or facts in its world model. In this sense, symbolic AI in the 1960s–1970s was very explicitly about connecting syntax to semantics: parse the sentence, then **ground** it in a formal representation of meaning (often logic). Winograd's approach involved translating English into a logical form that could be used to query the system's knowledge. This reflects the influence of people like Frege/Russell in AI – the idea that a natural language sentence can (and should) be mapped to logic (predicate calculus), at which point the *semantics* is clear (logic has a well-defined semantics in terms of models/world).

However, many early systems were brittle, partly because capturing *all* relevant semantics in explicit rules proved extremely difficult. Humans have a vast, implicit background of common sense knowledge. Attempts to catalog it (like Doug Lenat's Cyc project, an enormous ontology of common sense facts begun in the 1980s) showed just how much "semantic stuff" is needed to truly understand language or reasoning. This realization – that semantics (meaning, understanding of the world) is **huge and hard to formalize** – led to some soul-searching in AI.

Chomsky and the Primacy of Syntax

Parallel to these developments in AI, **Noam Chomsky** revolutionized linguistics in the 1950s with *syntactic* theory. In his 1957 book *Syntactic Structures*, Chomsky argued that syntax can be studied as an autonomous system – that is, the ability of humans to generate grammatical sentences follows rules that are independent of meaning. To illustrate this, Chomsky provided the famous example: *"Colorless green ideas sleep furiously."* This sentence is **syntactically well-formed** (adjective–adjective–noun–verb–adverb) but semantically nonsensical [86]. He contrasted it with a jumbled version "Furiously sleep ideas green colorless," which is not grammatical [86]. The point: *grammaticality is not the same as meaningfulness*. From this Chomsky concluded, *"the notion 'grammatical' cannot be identified with 'meaningful'… in any semantic sense."* [86]. This was a direct statement that syntax deserves separate study – one can have a perfectly correct sentence that means nothing, so syntax must be a different component from semantics.

Chomsky's theory introduced concepts like **deep structure** and **surface structure** and transformations between them. Early generative grammar focused on syntax, aiming to model the innate grammatical knowledge that allows humans to produce and understand novel sentences. Chomsky was less concerned (at least in the early years) with formal semantics; he viewed language as primarily a formal system *in the mind*. He even criticized attempts to base linguistics on behaviorist notions of "meaning" or statistical co-

occurrence, emphasizing instead an abstract computational system (Universal Grammar) that generates syntax.

However, Chomsky did not deny semantics – he recognized that a full theory of language needs a semantics component – but in practice, the Chomskyan revolution led to a huge focus on **formal syntax**, with semantics somewhat on the backburner until later. In the late 1960s and 1970s, researchers like Montague (who famously said "there is no important theoretical difference between natural languages and the formal languages of logicians") brought formal semantics into linguistics, showing how to interpret syntactic structures in model-theoretic terms. Montague's work essentially combined Chomskyan syntax with Fregean semantics, allowing one to derive the meaning of a sentence from its parse tree by applying semantic rules (e.g., the meaning of a noun phrase and a verb phrase combine to give the meaning of the sentence).

For AI, Chomsky's emphasis on an internal grammar reinforced the idea that language understanding involves two stages: first **parsing** (handling syntax), then **semantic interpretation**. Many natural language processing systems were built that way: e.g., a syntactic parser to get structure, and then a semantic interpreter to map that structure to some representation (like logic or database queries). This pipeline aligns with the intuition that syntax is the *vehicle* and semantics is the *content*. Any AI that aimed to truly understand language had to solve both: parse the sentence correctly (otherwise you might mis-assign "who did what to whom"), and have a knowledge base to link words to referents or concepts (otherwise the symbols remain meaningless). For instance, if the input is "the dog chased the cat," a parser would identify the subject as "the dog," the verb as "chased," object "the cat" – that's syntax. The semantic stage would then perhaps assert CHASE(Dog, Cat) in some knowledge representation, and maybe check or use facts like dogs are animals, chasing implies movement, etc.

Symbolic AI and the Search for Semantics

From roughly the 1950s to late 1980s, **symbolic AI** was the dominant paradigm. It assumed that intelligence could be achieved by manipulating *symbols* that represent things in the world, following rules. Here, *syntax* corresponds to the formal rules of symbol manipulation (like inference rules, production rules, or search algorithms), and *semantics* is provided by the human designers: each symbol was intended to mean something. For example, an expert system might have a rule IF "X is a bird" THEN "X can fly" (with exceptions). The symbols "bird" and "fly" refer to concepts in the world; the system manipulates these symbols logically. The **meaning** of the symbols isn't derived by the machine from experience – it's built in by us (we *know* what birds are and we program that knowledge in).

This approach met with successes in constrained domains (like medical diagnosis systems, or algebra problem solvers). But it also met famous challenges. One is the **frame problem** (how to handle the vast background knowledge and keep track of what changes and what stays the same after actions – a problem of pragmatic semantics). Another is the **commonsense knowledge problem** – to truly understand everyday situations, an AI needs an enormous amount of trivial knowledge (e.g., water is wet, people have feelings, gravity makes things fall, etc.). Encoding all that as formal symbols is daunting. And if the AI encounters a situation outside its coded knowledge, it fails to "understand." Symbolic AI systems were often brittle in that way: they didn't gracefully handle novel inputs that weren't anticipated by their creators.

By the late 1970s and 1980s, some researchers became disillusioned with purely symbolic approaches. This fomented the rise of **connectionism** – a paradigm inspired by how brains might work, using artificial neural

networks (ANNs) instead of explicit symbols. Connectionist models handle information in a distributed, numerical form (weighted connections, activations) rather than discrete symbolic representations.

The Connectionist Turn: Learning Syntax and Semantics from Data

Early connectionist work (like the Perceptron in the 60s, which was limited, and the resurgence in the 80s with multi-layer networks or "Parallel Distributed Processing" by Rumelhart & McClelland) offered a fundamentally different view of syntax and semantics. Instead of having human-crafted syntax rules and semantic representations, connectionist networks would **learn** from examples. For instance, in 1986 Rumelhart and McClelland famously made a network learn the past tense forms of English verbs. They showed it could generalize regular forms ("jump"->"jumped") and even handle some irregulars after training, without being explicitly taught grammatical rules – it statistically picked up patterns. This suggested that *some aspects of language could be learned as patterns rather than explicitly coded rules*. It was a shift towards **inductive learning of form**.

However, connectionism sparked a heated debate. Classical AI folks (and some philosophers) argued that neural nets, being just big numeric matrices adjusting weights, might capture *associations* but not the true **combinatorial, structured nature** of language and thought. Jerry Fodor and Zenon Pylyshyn, in a 1988 paper, argued that connectionist models **lack "systematicity" and "compositionality"** – hallmarks of human language and cognition [87] [88]. *Systematicity* means if you understand "John loves Mary," you can understand "Mary loves John"; our cognitive abilities come in coherent packages. Fodor and Pylyshyn claimed neural nets can't guarantee such systematic generalization unless they implicitly implement a classical (symbolic) architecture [87] [89]. Essentially, they were saying: **without explicit syntax, neural nets will fail to handle novel combinations** the way humans do. They believed the **Language of Thought** hypothesis (that thinking involves a mental syntax and semantics) underlies our capacity for systematic thought, and connectionist nets either have to smuggle in symbols or they won't match human abilities [87] [89].

Connectionists responded by creating more sophisticated network models that showed some combinatorial abilities (like recursive neural networks that could handle simple grammar, or networks that learned representations of structured data). Over time, especially in the 2000s and 2010s, these networks grew in power. We saw models like **word embeddings** that captured semantic relationships by training on large text – a very connectionist, usage-based approach to semantics (aligning with Wittgenstein's "meaning is use" in a sense, since usage patterns determined vector positions). We also saw **recurrent neural networks (RNNs)** and later **Transformers** that could process sequences of words and output translations, summaries, etc., without an explicit symbolic grammar in the traditional sense. These systems learned *implicitly* the syntax and some semantics from data.

This deep learning paradigm (modern connectionism) thus represents a full-circle moment: earlier symbolic AI insisted on explicit syntax rules and semantic models; now neural nets try to *absorb* syntax and semantics from example data. The success of machine translation, for instance, shifted from rule-based parsers with bilingual grammars (symbolic) to purely statistical or neural methods that infer translation mappings by reading millions of sentence pairs. By the 2020s, **deep learning models (like GPT)** could generate language that *respects* English syntax almost perfectly, despite never being given grammar rules – it's all induced. They also display *semantic* understanding in some tasks (like question-answering), but as discussed, whether this is genuine or just very convincing mimicry is debated.

From a historical view:

- **Symbolic AI (1950s–80s)** treated **syntax** and **semantics** as things to be explicitly designed. Syntax was often hardcoded (grammars), and semantics was encoded in knowledge bases. The plus side was clear interpretation – one could point to a symbol and say "this stands for X in the world." The downside was brittleness and limited learning.
- **Connectionism (80s–90s)** sought to collapse the distinction: in a network, there isn't a clear line between syntax and semantics; everything is just vectors and weights. But typically, these networks had difficulty with complex structure initially. They excelled at perceptual tasks but struggled to do what a symbolic reasoning system could. For example, understanding a logical sentence like "All A are B; X is A; therefore X is B" was hard for early neural nets, which are better at fuzzy pattern matching than strict logical inference.
- **Hybrid Approaches**: Some researchers advocated combining the two – e.g., neural networks for perception or low-level parsing, but symbolic reasoning for high-level logic. The field of **natural language processing (NLP)** in the 90s and early 2000s used a lot of statistical methods (precursors to deep learning) but still often relied on some explicit linguistic structure (like part-of-speech taggers feeding into probabilistic context-free grammars, etc.). There was a gradual shift: from rule-based to statistical to neural purely.

Deep Learning Era: Rediscovering Semantics?

In the 2010s, deep learning's dramatic successes (speech recognition, image recognition, then NLP with word embeddings and Transformers) led AI researchers to double down on data-driven approaches. We saw **unprecedented handling of syntax**: e.g., Google's BERT (2018) can fill in missing words in a sentence by deeply understanding context – effectively showing knowledge of syntax (what part of speech fits) *and* semantics (what meaning fits the context). Transformer models explicitly model context using attention, which can be seen as dynamically figuring out syntactic/semantic relationships between words (e.g., in "the bird that the cat chased was flying," attention heads might connect "bird" with "was flying" and "cat" with "chased," implicitly recognizing the relative clause structure).

One might say that deep learning **operationalized a lot of semantics in a subsymbolic way**. For example, these models learn factual knowledge (Paris – France) by weight adjustments; they learn sentiment (positive/negative tone) by example; they even learn some rudimentary "common sense" (that if someone is holding a cup they likely intend to drink). But because it's all implicit, it's hard to know *what* they know and *when* they might fail. They don't have explicit *ontologies* or *logic* to fall back on; everything is encoded in continuous parameters.

Interestingly, some in the field are now exploring adding back a bit of explicit structure – like **neurosymbolic approaches** that combine neural nets with symbolic reasoning modules, aiming to get the best of both worlds (robust learning + clear reasoning). That's essentially a new chapter in the syntax-semantics saga: maybe purely statistical models are too opaque (we get issues like adversarial examples, or nonsense outputs), so integrating some symbolic **semantic knowledge or constraints** could help.

To highlight **key shifts**:

- In **philosophy**, Frege and Russell gave us a formal notion of semantics separate from syntax. Wittgenstein (later) and others (like ordinary language philosophers, Austin, etc.) reminded that meaning also depends on use and context, beyond formal structure.
- In **linguistics**, Chomsky separated syntax from semantics sharply, but later linguists reintegrated them through formal semantics and pragmatics.
- In **AI**, early systems attempted to encode semantics directly (with logic, rules) – an approach that was powerful in limited domains but didn't scale well to the ambiguity and breadth of real-world language. Then AI shifted to learning-based methods that largely ditched explicit semantics in favor of pattern recognition. We've gained fluency but at the cost of interpretability – we have models that *act* like they understand, but it's unclear how much they truly "know" the meaning of what they say. This brings us full circle to the philosophical questions: Is "understanding" just behaving as if you understand (the Turing Test)? Or is there an inner qualitative or functional state that a machine must have to *really* understand (Searle's view, intentionality, etc.)?

As of today, many AI researchers acknowledge **symbol grounding** as an open problem: how can we ensure a machine's symbols (or vector representations) actually *refer* to things in the world, rather than just to other symbols? Cognitive scientists like Stevan Harnad in 1990 phrased this explicitly: we have *symbolic AI* that can manipulate symbols but they ultimately get their meaning from us (like a Chinese–Chinese dictionary that you can't decipher unless you have an outside source of meaning) [53]. Harnad argued that connecting symbols to sensorimotor experience is crucial (e.g., an AI should learn "apple" by seeing/tasting apples, not just reading definitions). This echoes the **Robot Reply** we discussed: embodiment might provide the grounding missing in pure symbol systems or pure text-trained systems. Modern multimodal AI is a step in that direction (like image+text models grounding words in pixels), but true **interactive grounding** (learning by doing in the world) is still in early stages.

In conclusion, the journey of syntax and semantics has been one of oscillation between focusing on formal structures vs focusing on meaning and use. **Frege** gave us the blueprint of formal semantics (truth and reference). **Russell** refined our understanding of how syntax can disguise semantics (descriptions). **Wittgenstein** warned that meaning isn't just in formal correspondence, but in use. **Chomsky** showed that syntax has an autonomy and complexity that must be accounted for (we can't reduce grammar to a simple set of meaning mappings; it has its own abstract rules). **Turing** suggested that if you get the syntax/behavior right, semantics (or at least the ascription of semantics) will follow – a very influential idea for AI. **Early AI** tried to build machines that explicitly handle both syntax (parsing, rule-following) and semantics (knowledge bases), but hit limits. **Connectionism and deep learning** then attempted to absorb both into massive data-driven models, yielding systems with impressive *syntactic* facility and some *semantic* capacity, but also raising new questions about *the nature of the "understanding" they achieve*.

Through all these developments, one theme stands: **the synergy and tension between syntax and semantics**. We've learned that neither can be ignored. Pure syntax with no semantics (as Searle argued) is empty – it doesn't truly capture intelligence. But pure semantics with no structural rules would be unmoored – we need structure to combine meanings productively. AI continues to seek the right balance, perhaps ultimately aiming for systems that have **systematic syntactic competence, grounded in semantic experience**. Achieving that would mean building machines that not only form sentences correctly but also *attach* them to real meanings – essentially solving what philosophy of mind calls the problem of **intentionality** in artificial systems. Each historical contributor moved us a step toward that vision, even if the end goal is not yet reached.

References:

- Searle, J. (1980). *Minds, Brains, and Programs*. Behavioral and Brain Sciences, 3(3), 417-424. (Original paper introducing the Chinese Room argument)
- Cole, D. (2020). "The Chinese Room Argument", in *The Stanford Encyclopedia of Philosophy* [5] [37] [43].
- *Internet Encyclopedia of Philosophy*: "Chinese Room Argument" [3] [7] [18] (overview of the thought experiment and replies).
- Bender, E. & Koller, A. (2020). *Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data*. ACL 2020. [52]
- Titus, L. (2024). *Does ChatGPT have semantic understanding?* Cognitive Systems Research, 83, 101174 [55] [56].
- Chalmers, D. (2020). "GPT-3 and General Intelligence" (contribution to Philosophers On GPT-3) [50] [90].
- Rini, R. (2020). Quoted in Zimmermann, A., *GPT-3: Philosophy and Policy* (CITP) [51].
- Wittgenstein, L. (1953). *Philosophical Investigations*. (Emphasis on meaning as use) [80].
- Fodor, J. & Pylyshyn, Z. (1988). *Connectionism and Cognitive Architecture: A Critical Analysis*. Cognition, 28(1-2), 3-71 [87] [88].
- Chomsky, N. (1957). *Syntactic Structures*. (Introduced the autonomy of syntax; includes the "colorless green ideas" example) [86].
- Turing, A. (1950). *Computing Machinery and Intelligence*. Mind, 59(236), 433-460. [82].
- Frege, G. (1892). *On Sense and Reference*. (Foundation of semantic theory) [91] [92].
- Russell, B. (1905). *On Denoting*. (Theory of descriptions) [93] [94].
- Winograd, T. (1972). *Understanding Natural Language*. (SHRDLU system).
- Harnad, S. (1990). *The Symbol Grounding Problem*. Physica D, 42, 335-346.
- **(Additional citations are embedded in-text above in the format 【source†lines】.)** [11] [90]

---

[1] [2] [3] [6] [7] [8] [9] [10] [11] [12] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [40] [41] Chinese Room Argument | Internet Encyclopedia of Philosophy
https://iep.utm.edu/chinese-room-argument/

[4] [5] [13] [14] [15] [37] [38] [39] [42] [43] [44] [45] [46] [47] [48] [49] [82] [83] [84] [85] The Chinese Room Argument (Stanford Encyclopedia of Philosophy)
https://plato.stanford.edu/entries/chinese-room/

[50] [63] [64] Philosophers On GPT-3 (updated with replies by GPT-3) - Daily Nous
https://dailynous.com/2020/07/30/philosophers-gpt-3/

[51] [61] [62] [65] [66] [90] GPT-3 Raises Complex Questions for Philosophy and Policy - CITP Blog
https://blog.citp.princeton.edu/2020/08/13/gpt-3-raises-complex-questions-for-philosophy-and-policy/

[52] Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data - ACL Anthology
https://aclanthology.org/2020.acl-main.463/

[53] WATCH: Princeton Language and Intelligence Director Explains ...
https://ai.princeton.edu/news/2025/watch-princeton-language-and-intelligence-director-explains-why-large-language-models-are

[54] Stochastic parrot - Wikipedia
https://en.wikipedia.org/wiki/Stochastic_parrot

55 56 57 58 59 60 67 Does ChatGPT have semantic understanding? A problem with the statistics-of-occurrence strategy
https://philarchive.org/archive/TITDCH

68 69 73 74 75 76 77 78 79 80 81 91 92 93 94 Language, Philosophy of | Internet Encyclopedia of Philosophy
https://iep.utm.edu/lang-phi/

70 71 72 Theories of Meaning (Stanford Encyclopedia of Philosophy/Winter 2010 Edition)
https://plato.stanford.edu/archives/win2010/entries/meaning/

86 syntax - On colorless green ideas - Linguistics Stack Exchange
https://linguistics.stackexchange.com/questions/47816/on-colorless-green-ideas

87 88 89 Connectionist semantic systematicity - ScienceDirect
https://www.sciencedirect.com/science/article/abs/pii/S0010027708002837