Does ChatGPT understand text?

Rod Smith

10 November 2023

ABSTRACT

Startling semantic claims have been made about ChatGPT, OpenAI's online computer chatbot said to be trained on text "scraped" from the internet then to understand, party or fully, text requests online users type. Yet John Searle's famous Chinese room semantic analysis of the computer concludes that no computer could understand text. I analyze ChatGPT using Searle's semantic tools and discover that Searle sees computers as Turing machines and hence receive and internally manipulate text. But the science shows that computers receive, manipulate and store electrons. ChatGPT never gets users' text questions. It's never in a position to understand them. Yet ChatGPT contains elements which, used differently, could contribute to human-like understanding. Also, the internet is made of computers so no text is stored on the internet. The myth that ChatGPT is trained on or understands text underscores AI's fundamental misunderstanding of semantics. Founder Turing rejected thinking, the key semantic process, removing it from the scope of research. AI doesn't understand intelligence. This is knee-capping progress towards AGI. I examine the science of the sensory interface. This suggests semantic fundamentals of knowledge acquisition and indicates which ChatGPT elements could support thinking including understanding the meanings of the shapes of text.

Table of Contents

1. INTRODUCTION

Genuinely intelligent computers will revolutionize society. Although not without risk, such machines would offer great societal benefits including alleviating poverty and mitigating even reversing catastrophic climate change. Is ChatGPT genuinely intelligent?

ChatGPT is a recently released computer chatbot said to understand text requests which online users type, in the same sense humans understand text requests. If true, this heralds epochal change in various societal domains. But if not true, could a modified ChatGPT understand? And what modifications would be appropriate?

In late 2022, American AI research firm OpenAI released its online text chatbot, ChatGPT which quickly stunned many with its apparently human-like understanding of the requests, called "prompts", which online users typed. In 2023, OpenAI release an upgraded product called GPT-4, and in its 100-page "GPT-4 Technical Report" explained:

> "We report on development of GPT-4, a large-scale multimodal model which can accept image and text inputs."

Extensive media interest followed ChatGPT's 2022 release. OpenAI chief executive, Sam Altman, in very high demand, took a world tour of interviews, panel discussions and talks (most available on YouTube). Backer, Microsoft, reportedly initially invested 10 billion dollars in OpenAI. Google, Amazon, IBM, and others are investing billions in the underlying technology, the connectionist large language model (LLM).

On 20 October 2023, in an article titled,"OpenAI in Talks for Deal That Would Value Company at $80 Billion", the *New York Times* reported:

> "OpenAI is in talks to complete a deal which would value the company at $80 billion dollars or more, nearly triple its valuation less than six months ago, according to a person with knowledge of the discussions. The company would sell existing shares..."

Following the great interest in ChatGPT, popular explanations of ChatGPT have followed a quite narrow line. It's trained on troves of text "scraped" from the internet. Many, experts and the commentariat alike, say that from this training it learns the meanings of the text, and further, that this is why it understands, fully or partly, the text questions online users type. Section 2, below, "Claims about ChatGPT", provides quoted passages of experts and key media outlets.

The question of understanding text is typically considered a topic of the field of linguistic semantics. I[1] consider the following questions:

- Are the semantic claims about ChatGPT true?
- Is Meta CEO Mark Zuckerberg right when he says, "*It’s very clear ... we’ve had a big breakthrough*"[2]?
- If the semantic claims that ChatGPT learns from or understands text are false, could a modified ChatGPT understand?
- And if so, what might be appropriate modifications?

ChatGPT can make obvious mistakes that indicate lack of understanding, or at least a lack of full understanding. It can confidently make false statements. I asked it to explain a certain type of tort and to give example cases from a certain jurisdiction. It did an impressive job of briefly summarizing the law, but the cases it confidently cited were pure fiction. None actually existed. I didn't suppose this indicated faux-knowledge of a blowhard. ChatGPT isn't human. Rather, I concluded that it indicated a basic lack of understanding of the meaning of my typed request.

---

1 My background includes a decade of graduate Philosophy of AI research under a leading Turing scholar, AI-related data compression patents (US 6,414,610 and 5,748,955), and two decades of independent AI research funded by license fees from a large manufacturing-accounting-distribution software package I wrote then licensed to a US multinational from 1994 to 2015.

2 See for example, Mark Zuckerberg, YouTube video, "Mark Zuckerberg’s timeline for AGI: When will it arrive? | Lex Fridman Podcast Clips", 2023, m:s, 0:15.

Similar "hallucinations" have been quite often reported[3]. Some steps have been taken by OpenAI and others to try to reduce hallucination, but hallucinating now seems a fundamental characteristic of large language models[4] which renders them unsuitable as a dependable source of fact and reasoning.

Solving problems like hallucination would presumably have benefits. Human-like general machine intelligence could help mitigate even reverse catastrophic climate change. The genuinely intelligent machine with its ability to reduce poverty, address widespread pollution, and mitigate climate change seems to offer more hope than harm.

If ChatGPT is not intelligent, is seems important to know. If it ***is*** intelligent then we ought to give it some arms, legs, eyes, ears (and a few other things) and set it to work in the sensorimotor sense. If it needs to be modified in order to be intelligent, it actually seems urgent that we identify the modifications.

Yet ChatGPT, if it genuinely understands just text would offer great societal relief in a realm not often mentioned. Many disputes among people remain unresolved because of the very high cost and variable quality of legal representation. The genuinely intelligent machine would be the death knell of the human legal profession, and this would imply a more just society.

Semantics. Claims about understanding the meaning of text are typically considered to fall within the ambit of the field of linguistic semantics. ChatGPT is a computer application. The only notable prior semantic analysis of the computer is American philosopher John Searle's famous (1980) Chinese room argument (CRA) including thought experiment of the Chinese room.

The CRA assumes that computers receive and internally manipulate text. Searle:

> "A computer is by definition a device that manipulates formal symbols"[5] ;
> ...all that 'formal means here is that I can identify the symbols entirely by their shapes[6]; A digital computer ... manipulates symbols and does nothing else[7]; symbols, by definition, have no meaning (or interpretation, or semantics) ... except insofar as someone outside the system gives it to them."[8]

Searle is talking about *internal* manipulation. The CRA concludes that computers could never understand anything because the things they receive and manipulate, in and of themselves, are inherently meaningless. Any understanding of the text is in the mind of the intelligent observer, and hence is "observer-relative".

Searle's idea of internal manipulation of text comes from his assumption that computers are Turing machines:

---

3 See for example, Martin Keen, Master Inventor, IBM Technology YouTube video, "Why large language models hallucinate".

4 See for example, OpenAI (27 March 2023), "ChatGPT-4 Technical Report", OpenAI.com; Ira A. Fulton, Arizona State University School of Engineering YouTube video,'"LLM Limitations and Hallucinations", 25 June 2023; and Phaedra Boinodiris, IBM Technology, YouTube video, "Risks of Large Language Models", 14 April 2023.

5 John Searle, (2014), "What your computer can't know", in *The New York Review of books*, 9 October, 2014. https://nybooks.com/article/archives/2014/oct/09/what-your-computer-cant-know/

6 John Searle, (1980), "Minds, brains, and programs", in *The Behavioral and Brain Sciences*, (1980) 3, page 417.

7 John Searle, (1997), *The Mystery of Consciousness*, London, UK: Granta, page 9.

8 John Searle, (2002), "Artificial Intelligence and the Chinese Room: An Exchange [with Elhanan Motzkin]", in *The New York Review of Books*, Volume 36, page 45, quoted in John Preston and Mark Bishop (eds), *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence*, Clarendon Press, Oxford, 2002, page 35.

> "[A] a revolutionary change took place when Alan Turing invented the idea of a Turing machine. ... for practical purposes, the computer you buy in a store is a Turing machine. It manipulates symbols according to computational rules..."[9]

Turing machines operate by internally manipulating text characters[10]. Searle then argues about the semantics of text.

There are mainly four things wrong with this approach of Searle.

1. Computers aren't Turing machines.
2. Computers don't receive or manipulate text.
3. We understand external things (such as text on billboards), not our own inner neural pulses, so why condemn computers for not being able to do what we can't do?
4. Trying to explain inner semantics (such as that of thinking) with external semantics (such as that of text) is a category mistake.

Searle's analytical tools are legitimate. It's just that his assumptions, his premises, are not. I borrow some of his semantic concepts and use them in seeking to answering the two questions, Does ChatGPT really understand the text questions online users type? But if not, could a *modified* ChatGPT understand?

In performing this investigation I examine the science of computer electronics.

I reason that ChatGPT does not understand text at present but does contain elements which could be used semantically. And that with various (fairly extensive) modifications, ChatGPT could understand text in the same sense we do. I suggest a semantic theory of knowledge acquisition then explain, using C and assembler code, how the core semantic element of this theory can be realized in a computer.

2. CLAIMS ABOUT CHATGPT

The two main claims made about ChatGPT are: (a) it's trained on text scraped from the internet, and (b) as a result of this training, it does or might understand, partly or fully, the meaning of text questions online users type.

2.1 That ChatGPT is trained on text

New York Times, 5 December 2022:

> "[ChatGPT is] trained on billions of examples of text pulled from all over the internet."

Washington Post, 28 December 2022:

> "[ChatGPT] was trained on a trove of internet text..."

---

9 John Searle, (2014), “What Your Computer Can’t Know”, in *The New York Review of Books*, 9 October 2014.

10 Alan Turing, (1936), "On computable cumbers, with an application to the entscheidungsproblem", in *Proceedings of the London Mathematical Society*, 2(42), (published 1937), 230-265, page 231.

Wired, 19 May 2023:

> "The browsing [of the internet by ChatGPT] seems to be limited to just text-based information on webpages..."

Forbes, 22 March 2023:

> "[ChatGPT] is trained in vast reams of information – articles, textbooks, the internet..."

Scientific American, 28 December 2022:

> "[ChatGPT] was trained on a vast corpus of human writing available online..."

The Guardian, 5 December 2022:

> "[ChatGPT] is trained on a huge sample of text taken from the internet."

BBC, 7 December 2022:

> "[ChatGPT] is trained on vast databases of text scraped from the internet..."

Techradar.com, 15 March 2023, online:

> "ChatGPT's most original GPT-3.5 model was trained on 570GB of text data from the internet, which OpenAI says included books, articles, websites, and even social media. [ChatGPT has] been trained on hundreds of billions of words..."

OpenAI.com, developer and supplier of ChatGPT:

> "ChatGPT is fine-tuned from GPT-3.5, a large language model trained to produce text. ... These models were trained on vast amounts of data from the internet written by humans..."[11]

Sam Altman, OpenAI chief executive:

> "[ChatGPT works] by ingesting a huge amount of text, a significant fraction of the internet."[12]

Andrej Karpathy, OpenAI Co-founder:

> "[ChatGPT] understand[s] a lot about the structure of the text [that it is trained on] and all the different concepts therein ... The New York Times ... trained a small GPT on Shakespeare. So you have a small snippet of Shakespeare and they trained a GPT on it."[13]

Geoffrey Hinton, "The Godfather of AI":

> "...from the data [from its neural network, ChatGPT] figures out how to extract the meaning of the [user's] sentence, and it uses the meaning of the sentence to predict the next word [of its text answer]. It really does understand..."[14]

---

[11] OpenAI.com, (2023), "What is ChatGPT", online FAQ retrieved 30 August 2023. Various documents are available at OpenAI.com. These are carefully worded, and somewhat strangely, quite scientifically sparse. However, the many YouTube videos of senior OpenAI officers including interviews and talks of chief executive, Sam Altman, during his world tour are more informative.

[12] Sam Altman, (2023), American ABC News 2023 interview, YouTube video, "Inside ChatGPT technology".

[13] Andrej Karpathy, (2023), Microsoft 2023 YouTube video, "State of ChatGPT".

[14] Geoffrey Hinton, YouTube video, "The Godfather in Conversation, Why Geoffrey Hinton is worried about the future of AI", University of Toronto", m:s. 26:15.

On Geoffrey Hinton's account, ChatGPT understands the meaning of the text answers it generates.

2.2 That ChatGPT understands text

Understanding text includes having knowledge, since understanding text presupposes having the knowledge of the meanings of the shapes of the text.

The Guardian, 5 December 2022:

> "ChatGPT is [an] AI chatbot capable of understanding natural human language..."

New York Times, 5 December 2022:

> "[ChatGPT's] knowledge is restricted to things it learned before 2021..."

OpenAI.com, "Introducing ChatGPT", OpenAI.com:

> "[ChatGPT's] ideal answer depends on what the model knows rather than what the human demonstrator knows."

OpenAI.com, "What is ChatGPT", online FAQ, retrieved 30 August 2023:

> "[ChatGPT] has limited knowledge of world [sic] and events after 2021..."

Medium.com, 3 December 2022, Colin Baird:

> “ChatGPT is trained on an incredibly large body of text data, which allows it to understand the context and meaning of words and phrases...”

Mira Murati, OpenAI Chief Technology officer:

> "[ChatGPT] can tell you if it doesn't understand a question and needs to follow up..."[15]

Sam Altman, OpenAI chief executive:

> "[ChatGPT works] by ingesting a huge amount of text, a significant fraction of the internet. This AI system can learn the underlying representations of what these words mean..."[16]

Andrej Karpathy, OpenAI Co-founder:

> "[ChatGPT] understand[s] a lot about the structure of the text [that it is trained on] and all the different concepts therein."[17]

ChatGPT, as explained by ChatGPT:

> "ChatGPT is a computer program that uses artificial intelligence (AI) to understand and respond to natural language text, just like a human would. It can answer questions, write sentences, and even have a conversation with you. It's like having your own personal robot that can understand and talk to you."[18]

---

15 Mira Murati, (2022), *The Washington Post*, 10 December 2022.

16 Sam Altman, (2023), YouTube video, American ABC News 2023 interview, "Inside ChatGPT technology".

17 Andrej Karpathy, (2023), Microsoft 2023 YouTube video, "State of ChatGPT".

18 A reply by ChatGPT, quoted in USA Today, (27 January 2023), "What is ChatGPT? Everything to know about OpenAI's free AI essay writer and how it works".

Geoffrey Hinton, "The Godfather of AI":

> "ChatGPT and GPT-4 ... know much much more than any one person. ... [They] know about 1,000 times more common-sense facts than we do. ... So maybe these things are much, much, better at acquiring knowledge. ... We've built something better [than the human brain] .... Large language models [such as ChatGPT] are understanding what was said..."[19]

Greg Brockman, Co-founder, President and Chairman of OpenAI:[20]

> "...we found a single neuron in that [LLM] model that had learned a state of the art sentiment analysis classifier. It can tell you whether it's a positive review or negative review [of an academic paper]. That's understanding, you know. I don't understand what "understanding" means, but it's semantics for sure...
>
> (11:45) "... so you can just literally upload a file [to ChatGPT] and ask questions about it, and very helpfully, you know, it knows the name of the file. And it's like, oh, this is CSV, a comma separated value file. I'll parse it for you.
>
> "The only information here is the name of the file, the column names like you saw, and then the actual data. And from that it's able to infer what these columns actually mean. Like that semantic information wasn't in [the upload]. It has to sort of put together its world knowledge of knowing that, oh yeah, arXiv is a site where people submit papers...
>
> (16:00) "And that's honestly one of the reasons we released ChatGPT. Together I believe that we can achieve the OpenAI mission: ensuring that artificial general intelligence benefits all of humanity."

And not just shareholders.

Perhaps the following is a paradigmatic encapsulation of media views:

Forbes, 20 March 2023:

> "Large language models (LLMs) are a type of artificial intelligence (AI) system that's been trained on large amounts of text data. They can understand natural language and produce human-like responses to inputs. These models use advanced machine learning (ML) algorithms to understand and analyze the nuances of human speech, including syntax, semantics and context meanings."[21]

3. SEMANTICS

The quoted passages above from experts and key media outlets alike clearly indicate that a certain conceptual framework is being used to understand ChatGPT and large language models generally.

I want to question this framework. Is it accurate? Is it science? To do this I borrow analytical tools from Searle's Chinese room semantic analysis of the computer. On examination, this is an analysis of the 1936 theoretical Turing machine rather than of the

---

[19] Geoffrey Hinton, (2023), YouTube video, "Geoff Hinton, the 'Godfather of AI', quits Google to warn of AI risks".

[20] Greg Brockman, YouTube video,"OpenAI's Greg Brockman: The Future of LLMs, Foundation & Generative Models".

[21] Forbes, (2023), "Beyond chatbots: The rise of large language models", 20 March 2023, at forbes.com.

electronic computer. But Searle's analytical tools are appropriate and yield important results when used to investigate actual computers.

In other words, as an explanation of the semantics of computer electronics, Searle's Chinese room analysis is wrong, but his analytical tools are fine. I use them (plus a few others) to examine the semantics of the science, the physics and chemistry, the electronics, the "electronic substrate" of actual computers. That's because what computers internally manipulate and store is inside the electronics.

3.1 Field of analysis

ChatGPT is a computer application said to understand the meanings of text. Understanding the meaning of text is typically considered a topic of the field of linguistic semantics.

3.2 Subjects of analysis

Chatbots are computer applications, or more fully, application of the electronic device now known as the "stored-program digital computer" ("computer").

Online humans use ChatGPT by typing text questions, called "prompts". This is typically done on a laptop or smart device. Then users sees text responses display on their screens.

The semantics of this situation encompasses these aspects:

- the human user, since they type the text,
- the user's local device, since this handles transmissions to and from ChatGPT,
- the communication channel itself, including the internet, and
- ChatGPT.

As for the process of understanding, the issue is partly whether ChatGPT understands the text which a user types, but it's also whether ChatGPT has the *same* understanding, qualitatively speaking, as was in the mind of the user when the user was typing.

3.3 Scope of semantics in AI

John Searle's 1980 Chinese room argument (CRA) is a semantic attack on Turing's (1950) Turing test for machine intelligence, and on AI's related claim that a computer could understand the meaning of text.

In linguistics, semantics typically concerns the understanding of the meanings of words including text. But the CRA circumscribes a broader scope. Searle first says:

> "A computer is by definition a device that [internally] manipulates formal symbols"[22]; ...all that 'formal' means here is that I can identify the symbols entirely by their shapes[23] [and] symbols, by definition, have no meaning (or interpretation, or semantics) ... except insofar as someone outside the system gives it to them."[24]

---

[22] John R. Searle, (2014, 9 October), "What your computer can't know", in *The New York Review of books.*

[23] John R. Searle, (1980), "Minds, brains, and programs", in *The Behavioral and Brain Sciences*, (1980) 3, page 417.

[24] John R. Searle, (2002), "Artificial Intelligence and the Chinese Room: An Exchange [with Elhanan Motzkin]", in *The New York Review of Books*, Volume 36, page 45.

Here, Searle is talking about text. A computer by definition (according to Searle) operates by internally manipulating text.

His associated thought experiment of the Chinese room, intended as an easy-to-comprehend picture of the essential parts and processes of the electronic computer, is about internally manipulating text, Chinese ideograms, and whether the man in the room (the computer CPU) understands the meanings of their shapes.

But Searle in his CRA then extends this idea of text as linguistically-interpretable shape to cover what computers receive from sensory apparatus. One reply (objection) to the CRA says that the Chinese room lacks human-like sensors (Searle 1980, p. 420), and that once these are added, the man in the room could understand the world.

In Searle's response to this objection, he is the man in the room (he is the computer CPU), and he receives and manipulates Chinese ideograms. These "formal symbols" come into the room from sensory apparatus (1980, p. 420):

> "Suppose, unknown to me, some of the Chinese symbols that come to me come from a television camera attached to the robot ... all I am doing is manipulating formal symbols ... All I do is follow formal instructions about manipulating formal symbols"[25]

And formal symbols in and of themselves are meaningless.

On this account, what the computer receives from television camera "eyes" are still text symbols. They are still things that Searle identifies "*entirely by their shapes*" (1980, p. 418), but they don't constitute words. Rather, their shape relates to the television camera's electronic image detection system. These things which the robot-controlling computer receives are still identified by shape, but the shapes have not been assigned a linguistic meaning.

3.4 The nature of understanding

Does ChatGPT understand text? This is not such an easy question to adequately answer. Understanding seems the hallmark of intelligence. This seems so, whether understanding the slipperiness of mud, the danger implied by the rustle of undergrowth typical of a predator, or the meanings of words including text. The process of understanding is not well understood with concepts also applicable to configuring computers.

Cases of understanding can be characterized as instances of a relationship between the inner and the outer. That is, between the brain and the environment. This is a relationship critical to survival and hence will have been selected for since quite early evolutionary times. Yet little is known about this primitive relationship of understanding expressed with concepts also applicable to understanding how to configure a machine to realize instances.

Though one machine, the electronic digital computer, seems to have sufficient processing speed and quantity of directly addressable storage to have human-like intelligence.

---

[25] John R. Searle, (1980), "Minds, brains, and programs", in *The Behavioral and Brain Sciences*, (1980) 3, page 420.

However it's not known how to configure this machine to survive in the wild. No machines, bar evolved organic ones, have been able to reliably do this so far. It used to be joked that robotic AI computer systems have less intelligence than an ant[26].

Humans understand external text. The text is in the environment. The understanding, by contrast, is internal in the brain. The text is outer, in books, on billboards, wherever. The understanding of the meanings of the text shapes is a process of the little-known mind. Yet one still wants to at least attempt an answer to the question, Does computer application ChatGPT understand text?

3.5 Conceptual frameworks used

I try to answer the question, Can ChatGPT understand text? Such an inquiry implies using a number of conceptual frameworks, or paradigms.

The obvious framework is that of semantics, since understanding is a topic of the field of semantics. But as this inquiry continues, other paradigms become very necessary. I examine a certain type of machine. Also relevant are concepts used to understand and describe the machine.

The chatbot at issue is a connectionist large language model. Concepts used to understand the connectionist theory of intelligence are relevant. ChatGPT is a computer application, so both hardware and software conceptual frameworks apply. This is the list of paradigms so far:

- semantics
- connectionism
- large language models
- computers

Then several conceptual frameworks are used to understand computers. These include:

- the Turing machine and computation
- the physics and chemistry of the electronics
- programming languages

This quite long list might not be as bad as it seems, given that the concepts are adequately explained. I try to do this when the concepts are first used, and also sometimes later.

3.5.1 Starting with semantics

The best place to start an inquiry into whether ChatGPT understands text is with the key topic of semantics. Given that ChatGPT is a computer application, this means an inquiry into the semantics of the computer.

Interestingly though, the idea of this place, the semantics of the computer, is an idea which the founder of the field of AI research and pioneer computer scientist, Alan Turing, said is almost completely incoherent. He founded the field on denying the very possibility

26 This claim is quite separate from the idea of "Simon's ant", an idea of Herbert A. Simon (*The Sciences of the Artificial*, 1969, MIT Press, p. 23, 52). The idea is that what seems complex survival-related behavior can be the case where the complexity is in the environment traversed by an organic system, not within the organic system itself which operates on simple rules.

of a machine having the principal semantic process, thinking. Understanding is a function of the process of thinking.

In his (1950) paper, "Computing Machinery and Intelligence", considered the manifesto, road map and founding treatise of the field of AI, Turing says (p. 442):

> "The original question, 'Can machines think?' I believe to be too meaningless to deserve discussion."

If so, there seems little point to considering whether a computer could understand anything. Thinking, the paradigmatic semantic process, the process which includes understanding language, cannot meaningfully be attributed to machines, Turing says.

Thinking as a concept certainly exists. Turing didn't deny that. In fact he uses it often. But he applies it to humans: "*I think it is unlikely*" (p. 435), "*we think it would be less likely*" (p. 443), "*most of us who think about it*" (p. 444), "*I do not think that*" (p. 444) and more. We use the term often, and people don't crinkle their brows in confusion. But according to founder Turing, trying to realize the process of thinking in a machine must fail. Yet thinking is the paradigmatic process of human intelligence.

We can consider this matter in more detail. In saying "*I believe [the idea of a thinking machine] to be too meaningless to deserve discussion*", Turing isn't claiming that the idea of a thinking machine is ambiguous. Ambiguous ideas still have meanings, just more than one. And he's not saying that the idea has an interpretation but no reference, or denotation. Names of fictional "entities" can still have clear meanings.

For example, we all know what unicorns are. Pony-like creatures with a tapering barley-twist horn protruding from the center of their foreheads. The idea is quite clear. The meaning is quite clear. It's just that, a far as we know, no such animals really exist. (But in a galaxy far, far away, who knows? They might, verificationism and the Vienna Circle aside.)

To say that a question is meaningless is a much stronger attack than alleging ambiguity or fictional reference.

Though if a meaning is an external thing, then to say that the term "thinking machine" is meaningless ***is*** to allege fictional reference. It's like saying that the term "unicorn" is meaningless because no such things exist. But if Turing is using the term "meaningless" in this externalist sense of lacking reference, it seems difficult to understand how the term "thinking machine" could be "*too*" meaningless. It seems unclear how existence or not could be a matter of degree.

In the present essay, a meaning is an inner semantic structure, and understanding is a process which operates in and between inner semantic structures.

In this case of a meaning being an inner semantic structure, to say a term or idea is meaningless is to say that no relevant inner structure exists. As regards text, it's to say that there is an external shape but no inner interpretation of it.

Turing founded the field of AI research explicitly on avoiding thinking. I want to argue that this was because Turing didn't understand semantics, and hence didn't understand either thinking or intelligence.

He didn't address semantics. If he understood it he would have addressed it. AI still doesn't understand semantics and hence still can't explain intelligence. The avoidance of semantics seems a main factor behind so much of the misunderstanding and so many of the false statements made by experts and popular opinion alike about ChatGPT.

3.5.2 Semantic concepts

I'd like to briefly outline main semantic concepts used later. Searle uses the concept of **symbol**. By "symbol" he means text character, the sort of thing Turing machines internally manipulate. The sort of thing the Chinese room internally manipulates. As Turing said about the Turing machine (1936, p. 231), "*At any moment there is just one square, say the r-th, bearing the symbol G(r) which is 'in the machine'*". The key phrase being "*in the machine*". Turin's second 1936 example machine manipulates inner text shaped "0" and "1", that is, numerals, digits. (The digits are the numerals shaped "0" to "9".)

With text, what has meaning, or interpretation, is the **shape** of the text. Shape is an **inherent property**, one not easily conceived of as existing separately from substance which bears it. An instance of a property is called a **value** of the property. A certain shape is a value of the property of shape.

Symbols (text) have an **extrinsic semantics** but no **intrinsic semantics**. The idea of extrinsic semantics is that the text itself doesn't contain, carry or indicate its meaning. Rather, a separate entity, an observer, reacts to the text shape and this reaction includes interpreting the meaning of the shape. Searle (2014) talking about the things Turing machines (he says "computers") internally manipulate:

> "symbols, by definition, have no meaning (or interpretation, or semantics) ... except insofar as someone outside the system gives it to them."

In other words, the meanings of the shapes of text are **observer-relative**.

Searle also characterizes text as **syntactic**, or **formal**. Syntax, or formality, can be thought of as the form, or shape, of the text. In Searle's sense, syntax is contrasted with and is opposed to semantics.

Reference. Text shape indicates nothing about the meaning of the shape or the reference of the shape. The shape "Eiffel Tower" gives no indication of what is understood by the shape (its meaning), or of the tall metal structure in Paris, France, going by that name (its reference). Similarly, the shape "Taj Mahal", gives no indication of what is understood by the shape, or of the marble building in Agra, India, going by that shape.

Searle uses the concept of **intentionality**, or aboutness. Intentionality is a property of mental states such as fear, belief and desire. These states are *about* particular things or types of thing. For example, spiders, Saturn and chocolate.

Knowledge has **semantic content**. Understanding text presupposes knowledge including knowledge of the meaning of the shapes of the text. **Understanding** is a process which operates within knowledge. Knowledge has an intrinsic semantics, it has semantic content. To have semantic content is to have an intrinsic semantics. Elements of knowledge have semantic content. By contrast, a token (unit of substance) is **semantically vacant** if it lacks both an intrinsic and an extrinsic semantics.

The **outer-to-inner interface**. I describe some things as **inner** or **outer**. The inner is the realm between the inner transmitting side of sensory apparatus and the brain, or central system, including the central system itself. The outer is the realm beyond the detecting surface of the sensor, in other words, the environment.

3.5.3 Importance of the Chinese room argument

As far as I know, the first and only prior semantic analysis of the computer is the famous semantic attack against the possibility of computer intelligence now known as the Chinese room argument (CRA). In this, author John Searle (1980) assumes that computers operate by internally manipulating text, which he calls "symbols".

I think the CRA is a crucial step on the path to machine general intelligence (AGI), and further, that the associated thought experiment of the Chinese room is probably one of the most important thought experiments of science. I mean this in Thomas Kuhn's sense where the thought experiment crystallizes a crisis of science[27]. In other words, AI is in a crisis of science, and the Chinese room thought experiment reveals what AI needs to solve.

It needs to do this because the CRA is a strong semantic argument about the computer, the only machine with seemingly sufficient inner speed of state change and sufficient uniquely identifiable storage locations to have human-like intelligence.

Searle's thought experiment uses the current understanding of the computer and attacks the claim that a computer, as so understood, could have human-level intelligence. Hence it is a classic thought experiment of science which inspires a search to understand the computer (so-called) with new and better ideas.

Review. The research field, "Artificial Intelligence", was established on an entreaty to avoid the semantic foundation of intelligence, thinking. Turing (1950, p. 442), "*I believe [the idea of a thinking machine] to be too meaningless to deserve discussion*". Thus the field was founded on the principle of denying the possibility of machine intelligence.

Having denied the possibility in a machine of the inner process of thinking, Turing sought to redefine the term "intelligence" to mean behavior (1950, p. 442):

> "I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted."

That is, that the term "thinking" will come to be understood to mean exhibiting a certain sort of behavior, as per passing the Turing test. It's little wonder, then, that current AI doesn't understand intelligence. Founder Turing explicitly excluded it from the scope of research.

---

[27] See for example, Thomas S. Kuhn, (1962), *The Structure of Scientific Revolutions*, The University of Chicago Press.

3.5.4 The CRA's two conclusions

The CRA makes a key mistake. The argument has two main conclusions. The first about computation, the process. The second about the computer, the programmed hardware, the electronic device. Searle's big mistake was to impute the first conclusion about computation to the electronic device.

The Chinese room argument is fundamentally about computation, the process, the process Turing was talking about when he said (1936, p. 249): "*Computing is normally done by [a human] writing certain symbols on paper*".

Of course, if computers did noting else but execute this process, that would be fine. But Searle never discusses the electronics. He doesn't address whether what semantically happens inside the text-manipulating Turing machine also happens inside computer electronics. As will be seen, computer electronics have a very different semantics.

In short, Searle says that his second conclusion, the one so devastating to AI (that computers could never possibly understand anything), is about the electronic hardware, but it's not. It's about computation. Put another way, Searle's conclusion is about some machine, just not the one AI is trying to make intelligent.

If true, this would seem a great relief to AI research, releasing it from the specter that the CRA's second conclusion might actually be true.

3.5.5 Concepts related to electronics

In my analysis, I understand the computer with the science of the physics and chemistry of the machine's electronic substrate. Related concepts include: *electromagnetic radiation (visual and radio spectra),* the *electron*, *electron field effects*, *electrical current*, *potential difference, voltage*, *buses*, *modules* (*integrated circuits, chips*), *PCB (printed circuit board)*, *capacitors*, *diodes*, *transistors*, *memory cell*, *charged trap flash*, *V-NAND*, *DRAM*, *VRAM* and *doped silicon*.

I also adopt electronic concepts including *binary difference, electronic substrate* (*semiconductor substrate*), *binary difference in motion*, and *binary difference at rest*. A key concept is that of electronic *sensory transduction*.

I also use the older established ideas of the *5-unit* (*5-element*) *code, mark, space, sending station, receiving station, figures (FIGS), letters (LTRS)*, *symbols* (%, &, $...), *mercury delay line*, and *ultrasonic pulse encoding*.

Computer programming concepts I use include: *register*, *direct memory addressing*, *pointer, indirection, shift, jump, call, procedure, record structure, field, absolute address, offset address* and *relative address*.

3.6 Historical background

AI research is characterized by attempts to realize in a computer two alternative and fundamentally different conceptions of intelligence. These are the symbolic paradigm and the connectionist paradigm.

The symbolic paradigm is based on the idea that computers are practical versions of Turing machines and hence operate by internally manipulating text symbols, as do Turing machines (Turing, 1936, p. 231). The connectionist paradigm is based on brain-like structures known as artificial neural networks (ANNs).

Celebrated Canadian AI researcher Geoffrey Hinton, often nowadays introduced as "The Godfather of AI"[28], explains the difference:

> "There are two different models of what intelligence is all about. ... [One] went with the idea that the knowledge you store is symbolic expressions. ... [I]nside your head is something a bit like sentences but cleaned up. And there's a completely different model of intelligence, which is that it's all about learning the connection strengths in a network of brain cells..."[29]

Perhaps enigmatically, researchers seeks to realize both models in the same device – the computer. This raises the question, If computation is symbolic (and it is) then how could non-symbolic connectionist theory be realized in a computer? Hinton offers a common answer: "*...neural net[s are] simulated on digital computers...*"[30].

The idea of **computer simulation**, derivative in part from the Church-Turing Thesis[31], is that any machine or natural system ("target system") which can be quite precisely described can be simulated in a computer. The description, text, is the computer program[32].

The August 1955 flier for the summer workshop which formally introduced the title "Artificial Intelligence" in America explained:

> "We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it."

The idea of computer simulation is that the computer receives input text symbols. These describe an input which the target system could receive. The computer then consults the program, a description of the target system typically in the form of many conditionals (if-then's). The computer then emits output symbols. These describe the output the target system would have produced had it actually received the input described by the computer's input symbols.

Early dominance of symbolic AI. Early AI researchers, Allen Newell and Herbert A. Simon, attended the 1956 Dartmouth College summer workshop and introduced their Logic Theorist symbolic AI program. This was said to deduce mathematical proofs

---

[28] Incorrectly introduced. Hinton was foundational in 1970s connectionist deep learning theory and practice. But the researcher who presented the AI baby at baptism and spoke on its behalf was the founder of symbolic AI research, Alan Turing. Media, nowadays, seems to be expunging symbolic AI from the popular consciousness. Which seems unfortunate since it suggests repetition of past mistakes.

[29] YouTube video, "The Godfather in Conversation, Why Geoffrey Hinton is worried about the future of AI", m:s. 4:50.

[30] YouTube video, "The Godfather in Conversation, Why Geoffrey Hinton is worried about the future of AI", m:s. 2:40.

[31] See (Turing, 1936) for an explanation of the universal Turing machine; and Alonzo Church, (1936), "An unsolvable problem of elementary number theory", in *American Journal of Mathematics*, vol. 58 (2), pages 345 – 363, presented to the American Mathematical Society, 19 April 1935.

[32] This idea, though not under the name "simulation", was introduced by Turing in his (1936) paper (p. 240-241) when he explains that a description ("standard description" , or "S.D.") of a target system, the simple Turing machine which prints "0"s and "1"s, can be loaded on the tape of another Turing machine configured as "universal". The universal machine then operates on this description text and emits the output which the target machine would have emitted.

including 38 of the first 52 proofs in the second chapter of Russell and Whitehead's legendary 1910 volume, *Principia Mathematica*.

The Logic Theorist and Newell and Simon's later General Problem Solver programs inspired many in early AI days. For the following half century, the symbolic paradigm dominated much of AI research, while enthusiasm for the connectionist alternative based on simplified brain structure fluctuated and was relatively inconsequential at that time.

Other nascent theories of intelligence also suffered thanks to the dominance of the symbolic paradigm. Renaissance man, Norbert Wiener, in the 1961 Second Edition of his 1948, *Cybernetics*, referring to the widespread enthusiasm for the symbolic model, advised:

> "...it behoves the cyberneticist to move on to new fields and to transfer a large part of his attention to ideas which have arisen in the developments of the last decade..."[33]

Cybernetics was an inspiring amalgam of different perspectives developed from Wiener's wartime work on anti aircraft guns which fundamentally entailed the process of feedback. Cybernetics was an early entry on the list of now-largely-ignored AI theories.

Another which soon found its way onto the list was James Gibson's 1950 "ecological theory" of perception[34] developed from his wartime research into certain visual problems of airplane pilots. His theory focuses on the idea that perception is a process which "picks up" information which exists independently in the environment. Such "affordances" are information about an object's use which an animal may or may not make use of:

> "Because of illumination the animal can see things, because of sound it can hear things, because of diffusion it can smell things. The medium thus contains information..."[35]

Symbolic AI, in its turn, is now largely abandoned as a path to AI's original goal, machine general intelligence. Thus, symbolic AI itself is now on the list of largely abandoned theories. Though since the 1980s and still today some academicians seek to reconcile symbolic and connectionist models[36].

Amazement similar to that now attending ChatGPT followed release of the 1950's symbolic Logic Theorist. Given the checkered history of theories of intelligence, we might now ask, Is ChatGPT the breakthrough many such as Mark Zuckerberg[37] believe it to be, or will it suffer the same fate as The Logic Theorist, symbolic AI generally, and others?

---

33 Norbert Wiener, (1948), *Cybernetics*, 1961 Second Edition, MIT Press, page viii.

34 James Gibson, (1950), *The Perception of the Visual World*, Houghton Mifflin Company, Boston; also (1966), *The Senses Considered as Perceptual Systems*, Houghton Mifflin Company, Boston; and (1979), *The Ecological Approach to Visual Perception*, reprinted in 1986 , Lawrence Erlbaum Associates, New Jersey.

35 James Gibson, (1979), *The Ecological Approach to Visual Perception*, Lawrence Erlbaum Associates, New Jersey, 1986, page 17.

36 See for example, Steven Pinker and Jacques Mehler (eds) (1989), *Connections and Symbols*; and Stevan Harnad, (1990), "The Symbol Grounding Problem", in *Physica D 42: 335-346*. Available on-line at www.ecs.soton.ac.uk/~harnad/Papers/Harnad/ in November 2004; and Stevan Harnad. (1994), "Computation Is Just Interpretable Symbol Manipulation; Cognition Isn't", in *Minds and Machines*, Vol. 4, pages 379-302.

37 See for example, Mark Zuckerberg, YouTube video, "Mark Zuckerberg’s timeline for AGI: When will it arrive? | Lex Fridman Podcast Clips", 2023, m:s, 0:15.

I argue that current ChatGPT is not intelligent, but does contains elements which, if used differently, could contribute to understanding. It contains elements which could contribute to knowledge. But the elements at present are not used semantically.

If true, then ChatGPT hopefully won't suffer the fate of symbolic AI, cybernetics and ecological theory, since it is a significant practical step forward on the path to genuine human-like machine intelligence.

4. ARGUMENTS

In this section, "Arguments", I'd like to present the body of my arguments about ChatGPT and also present further arguments about AI generally.

The semantic claims of interest made about ChatGPT concern acquiring knowledge of the meanings of the shapes of text. Specifically, that ChatGPT:

(a) is trained on text scraped from the internet,
(b) from this training it learns the meanings of the text, and
(c) this is why it understands, partly or fully, the text questions, or "prompts", which online users type.

There are dependencies and assumptions between these claims. For example, if no text exists on the internet, then there is nothing there to be scraped. And if nothing is there to be scraped then there is no learning from scrapings, and if there is no learning from scrapings then there is no resulting understanding of the meanings of the shapes of the scrapings.

Happily, then, all this, initially at least, comes down to one question: *Does text exist on the internet?* If the answer is, *No*, then all three claims (a) – (c) are false. If the answer is, *Yes,* then we need to go to the next step and consider claim (b): from its training on internet text, does ChatGPT learn the meanings of the shapes of the text?

It only remains, then, at this point, to consider whether the claim (a), "*Text exists on the internet*" is true or false.

How might this determination be made?

The first thing, we can say, is to identify the nature of text. What is text? Once that is clarified, then it can be asked, "*What is the internet made of?*" and then, "*Is any of this stuff actually text?*".

4.1 What is text?

Text is written or printed language such as we find in textbooks. It comprises instances of shapes. But not every instance of a shape is text. A shape is text if it has been assigned a meaning, or interpretation. This is often done by a community.

Such shapes might be atomic such as A, B, C, D and 1, 2, 3, 4, that is, letters of an alphabet and digits. Or they might be such shapes sequenced into words, multi-digit numerals, and so on.

Types of text characters include the just-mentioned words and numerals, and also punctuation marks, special characters such as %, &, @, mathematical symbols, +, -, ^, =

and others. In short, the shapes one finds imprinted on the top surfaces of keys of a computer or typewriter keyboard.

To understand text is to know the meanings of its shapes. Understanding text presupposes learning the meanings of the shapes. This is what many enthusiasts think ChatGPT does. They say it's rained on text scraped from the internet and that this training results in ChatGPT learning the meanings of the text shapes.

4.2 Where is text stored?

The internet is made of computers. To determine whether the internet contains text, first we want to ask, Do computers contain text? In other words, What are computers made of and what can they store? And as regards the computers which comprise the internet, do any of these computers contain text?

But there is also the question of what we understand a computer to be. So first, how are we to understand the computer?

4.3 What are computers made of?

Three paradigms were suggested earlier for understanding the computer: the Turing machine and computation, the physics and chemistry of the electronics, and programming languages.

4.3.1 The Turing machine conception

In this section I'd first like to consider the Turing machine, what Turing in 1936, before electronics existed, called a "computing machine" (1936, p. 231). Turing machines internally manipulate text. Turing (1936, p. 249): "*Computing is normally done by [a human] writing certain symbols on paper*", and (p. 231), "*At any moment there is just one square, say the r-th, bearing the symbol G(r) which is 'in the machine*'".

Turing:

Electronic computers are essentially Turing machines (1947, 107):

> "Digital computing machines ... may be regarded as practical versions of [the universal Turing] machine."

Electronic computers manipulate text numerals (1951, 1):

> "The information stored on paper by the human computer will mostly consist of sequences of digits drawn from 0, 1, ..., 9. ... The number [however] for the Ferranti machine [the Manchester Mark II computer] is two, and the symbols used are 0 and 1."

Electronic computers manipulate text numerals (1950, 441):

> "Usually fairly lengthy operations can be done such as 'Multiply 3540675445 by 7076345687' but in some machines only very simple ones such as 'Write down 0' are possible."

Here, Turing says computers internally manipulate text as do Turing machines. The idea that computers are Turing machines, or essentially Turing machines, is very widespread:

Paul Churchland (1984):

> "...the modern computer is a universal Turing machine."[38]

Philip Johnson-Laird (1988):

> "[The Turing machine] is the abstract ancestor of the modern digital computer."[39]

Roger Penrose (1989):

> "[The universal Turing machine] is indeed remarkably well approximated by the electronic computers of today."[40]

David Pool et al (1998):

> "A Turing machine is an idealization of a digital computer..."[41]

Jack Copeland (2000):

> "All modern computers are in essence universal Turing machines."[42]

Stevan Harnad (2006):

> "...the "Turing Machine" [is] the abstract description of a computer."[43]

John Searle (2014):

> "...for practical purposes, the computer you buy in a store is a Turing machine."[44]

More generally, Turing scholar B. Jack Copeland:

> "In 1935 ... Turing conceived the modern computer. He described an abstract computing machine consisting of a limitless memory and a scanner that moves back and forth through the memory, symbol by symbol, reading what it finds and writing further symbols."[45]

And:

> "...[Turing] pioneered the theory of computation, introducing the famous abstract computing machines soon dubbed 'Turing machines' ... [His 1936]

---

38 Paul Churchland, (1984), *Matter and Consciousness: A Contemporary Introduction to the Philosophy of Mind*, page 105. According to Wikipedia, Paul Churchland is a Canadian philosopher specializing in philosophy of mind, neurophysiology and AI, and is currently Professor Emeritus at the University of California, San Diego.

39 Philip N. Johnson-Laird, (1988), *The Computer and the Mind*, Harvard University Press, Cambridge, Massachusetts, 1988, page 51. Philip Johnson-Laird is a philosopher specializing in language and reasoning, a member of the American Philosophical Society, Fellow of the Royal Society, Fellow of the British Academy, Fellow of the association for Psychological Science, and Fellow of the Cognitive Science Society. He has been awarded honorary doctorates from Göteborg, Padua, Madrid, Dublin, Ghent and Palermo. Yikes!

40 Roger Penrose, (1989), *The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics*, London, Vintage, page 48. Roger Penrose is a British mathematician, physicist, philosopher of science, Nobel Laureate in Physics, honorary fellow of St John's College, Cambridge, and of University College London, and currently emeritus fellow, Wadham College, Oxford.

41 David Poole et al, (1998), *Computational Intelligence: A Logical Approach*, page 4.

42 B. (Brian) Jack Copeland, (2000), "Alan Turing and the Origins of AI", online at www.alanturing.net. Jack Copeland is Professor of Philosophy at the University of Canterbury, Christchurch, New Zealand, specializing in logic and Alan Turing. My old university.

43 Stevan Harnad, (n.d.), "The Annotation game: On Turing (1950) on Computing, Machinery and Intelligence". http://www.ecs.soton.ac.uk/~harnad/Temp/turing.html. Stevan Harnad is a Canadian cognitive scientist based in Montreal.

44 John Searle, (2014), “What Your Computer Can’t Know”, in *The New York Review of Books*, 9 October 2014. John Searle is an American philosopher specializing in language, mind and intentionality.

45 B. Jack Copeland, (2000), "Alan Turing and the Origins of AI", online at www.alanturing.net.

> 'On Computable Numbers' is regarded as the founding publication of the modern science of computing..."[46]

Roger Penrose well expresses the established wisdom: "*[The universal Turing machine] is indeed remarkably well approximated by the electronic computers of today*".

That might be true in some respects but not in the semantic one. Semantically the two machines are different. What a Turing machine internally manipulates has an extrinsic semantics, a key premiss of the Chinese room argument which Searle wrongly applies to the electronic computer. But the instances of binary difference[47] internally manipulated by actual computers have no semantics at all. These instances are semantically vacant[48].

This semantic difference between Turing machines and computers is crucial because it puts to rest the myth that computers internally compute. Computation is manipulation of meaningful tokens, that is, text. Turing (1936, p. 249): "*Computing is normally done by [a human] writing certain symbols on paper*".

The computation "1+2=3" is true because of the meanings and sequence of the shapes, and "3+2=1" is false for the same reason. But the instances of binary difference electronic computers manipulate and store are semantically vacant. No linguistic meanings have been assigned to values of properties of the transmitted or stored electrons.

The established wisdom says that computers do contain text because they are practical versions of Turing machines. Turing (1947, p. 107), "*Digital computing machines ... may be regarded as practical versions of [the universal Turing] machine*". But this view is mistaken. They don't store or manipulate inner text. Hence we still need to understand the nature of the computer. We move on, then, to examine the science of the electronics.

4.3.2 The electronic conception

The second paradigm for understanding the computer is the science, the physics and chemistry, of the electronic substrate. If text is stored on the internet, it is stored in this substrate. So what, exactly, is computer storage made of?

4.3.3 What is computer storage made of?

Suppose I read the Wikipedia online article on ChatGPT. The article exists on Wikipedia's servers not as text but as electrons inside electronic substrate. This substrate contains all the Wikipedia articles. Certainly, what displays on my screen is text. But what is stored is electrons. The substrate is made of components. So what are these components and what, according to the physics and chemistry, happens inside them?

The short answer is that electrons are stored. These are stored in minute capacitors or alternatively, in microscopic electron "traps", such as those in the charged trap flash of V-RAM semiconductor storage.

---

46 B. Jack Copeland, (2004), *The essential Turing: The ideas that gave birth to the computer age,* Oxford, UK: OUP, page 6.

47 In modern semiconductor storage such as V-RAM, a "memory" cell can have more than two possible states. These are named as though they are binary states. For example, for the 4-state cell, the states are given the names "00", "01",, "10" and "11", and converted back into binary sates when needed. So the single 4-cell V-RAM state named "01" would be converted back into two actual binary state units, the first named "0" and the second named "1".

48 Like the electrical pulses in the human brain.

The capacitors are either charged or discharged (sates named "1" and "0"). In V-RAM flash "memory" the electrons are held between two dialectic layers, but not in the way typical of capacitors. Many good YouTube videos explain this in detail.

4.3.4 Does text exist in computer storage?

Anyone who builds or disassembles computers sees text. Meaningful shapes such as "Intel" are printed on top surfaces of modules (chips, integrated circuits). Shapes like "R23" are printed on PCBs (printed circuit boards). "R23" typically means resistor number 23 on the respective circuit diagram. In older computers, "100μF" might be seen on cylindrical component, capacitors, indicating a capacitance of 100 microfarads.

However, this text printed on components is irrelevant to whether text exists in computer storage. The machine can't create, delete, move, or react to this surface text. In short, it's not manipulable. It plays no part in the operation of the machine. It's simply printed on exposed surfaces of components so humans can see the shapes. Surface text is irrelevant to the machine itself.

So does text exist in computer storage? We know what text is: instances of substance whose shapes have been assigned meanings. Do the shapes of the microscopic groups of stored electrons have linguistic meanings? No. The shape of a group is actually irrelevant to what is stored. The binary difference is difference in voltage or quantity of electrons. Shape is entirely irrelevant.

So what conclusion must we reach? It seems inescapable. No text exists in computer storage.

Given that the Turing machine defines machine computation as manipulation of instances of linguistically meaningful shapes (text, including "Standard Descriptions" or "S.D."s (1936, p. 240)), the electronic "computer" (so-called) doesn't perform inner computations. Turing's 1950 paper about the electronic computer titled "Computing Machinery and Intelligence", might more accurately have been titled, "Non-computing Machinery and Intelligence"[49].

4.3.5 The relationship between computers and text

When I view a Wikipedia article, I see text displayed on my screen. I also see atomic text characters on the keys of my keyboard. I might print the article, in which case I see text on the paper which comes out of my printer. If I open up a computer case I see text printed on components, such as the company name "Intel".

That's all.

The only place text exists in this entire picture is on exposed surfaces of components or of peripheral attachments – keyboard, screen, printed paper. It occurs there *and nowhere else*. And just for one purpose: so humans can see the shapes, interpret their meanings, and use the machine *as a tool*.

---

[49] The title of my early-2000s doctoral thesis.

4.4. What is the internet made of?

Since the internet is made of computer storage, and since there is no text in computer storage, no text is stored on the internet. And since the only way text could get on the internet is by being stored there, no text exists on the internet.

The idea, then, that text exists on the internet is false. This myth also applied to images. There's no such thing as images stored on the internet. Or sounds, or smells, and so on. And for the same reason as there's no text.

We don't have images in our brains. Or text. Centuries ago erudite opinion abandoned the Aristotelian idea that recollections comprise something like inner images. If AI wants to understanding how a computer could have human-like intelligence, it needs to abandon myths which, when said of the human brain, were abandoned centuries ago. Such myths as that there are vast troves of text or images on the internet.

Descriptions of ChatGPT are very often misleading. We now know it's not trained on text from the internet because none exists there in the first place. It doesn't get users' text questions – all it gets is electrons. It's never in a position to understand users' text questions because it's never exposed to them. The only place the questions exist is on exposed surfaces of user devices.

4.5 AI's failure to understand semantics

The need to address semantics has been recognized since early AI days in some quarters. So how has AI approached this need?

MIT professor and lauded AI visionary **Marvin Minsky**, for example, knew that his popular 1968 book, *Semantic Information Processing*[50], didn't relate to semantics. This is discussed in some detail in the next section.

Newell and Simon's 1970s influential Physical Symbol System Hypothesis (PSSH) used the semantic ideas of external designation, denotation and reference but applied them to what is *inside* computers[51].

The two authors don't explain sense perception, the semantic relation between the inner and the outer by virtue of which we come to know the world. Their solution: in their PSSH theory of intelligence, they ignore sense perception:

> "We will simply take the letters of the English alphabet, the digits, punctuation marks, and a few special symbols ... as primitive symbols designating external stimuli".[52]

And:

> "...at the time we deal with them, stimuli and responses in the external environment will already be encoded internally – designated by [internal]

50 Marvin Minsky, (1968), *Semantic Information Processing*, Cambridge, MA: MIT Press.

51 Allen Newell and Herbert A. Simon, (1972), *Human Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall; Newell and Simon, H. A, (1976), 'Computer science as empirical inquiry: symbols and search ', in *Communications of the ACM*, 19(3), March 1976, pages 113-126. Delivered as the 1975 Turing Award Lecture, ACM Annual Conference, Minneapolis, 20 October 1975.

52 For instance, Allen Newell and Herbert Simon, (1972), *Human Problem Solving*, page 25.

> symbol structures. Hence we will almost always deal with these designations rather than with the external objects designated."[53]

Translation: we will always deal with these internal designations and never with the external objects designated. Hence Newell and Simon, both ACM Turing Award recipients[54], avoid the semantics of designation. The authors ignore sense perception, then confirm this, saying:

> "In taking the letters of the alphabet as primitive symbols in our analysis of problem solving, we divorce the problem of solving theory from detailed concern with the sensory mechanism."[55]

Translation: ...from any concern with the sensory mechanism. On the website of the project, Soar[56], which seeks to realize the Physical Symbol System Hypothesis, in 2005:

> "Soar does not yet have a standard model for low-level perception or motor control."[57]

Translation: or any model. This, in 2005, over 30 years after the PSSH was proposed. I confirmed this by email with the Soar project team. This avoidance for decades of sensory apparatus is quite typical of symbolic AI generally. It speaks to the lack of understanding of the core inner-outer semantic relation, and the absence of candid admissions of this important problem.

In the 1990s, roboticist **Rodney Brooks** claimed to escape the symbolic paradigm. But the gist of the symbolic paradigm is that humans define the causation of the AI system. Though Brooks talks of signals rather than symbols, the reactions of his "insects" and other robots are defined by human knowledge of the world[58]. Yet in a semantic system, most knowledge derives from the machine's own sensory reactions with the environment.

AI hasn't understood semantics. One response: take the term "semantic", since any theory of human-like intelligence will need to use the term "semantic", then apply it to things which either have no semantics, or have only an extrinsic semantics, that is, where respective semantic content is inside the mind of the intelligent observer.

For example, AI developed data structures which it calls "semantic nets". The semantics, it's quite clear, is inside the brain of the human observer, not inside the data structure. The AI applications called "expert systems" get their name from the human expert. The semantic content is inside the human. AI "knowledge bases" contain no actual knowledge. The only knowledge is inside the brain of the human observer of shapes displayed on screens or printed on sheets of paper.

---

53 Allen Newell and Herbert A. Simon, (1972), *Human Problem solving*, page 21.

54 The Preamble to Newell and Simon's 1975 popular paper, "Computer Science as Empirical Inquiry: Symbols and Search" starts, "The 1975 ACM [Association for Computing Machinery] Turing Award was presented jointly to Allen Newell and Herbert A. Simon at the ACM Annual Conference in Minneapolis, October 20".

55 Allen Newell and Herbert Simon, (1972), *Human Problem solving*, page 26.

56 "Soar", formally capitalized, "SOAR", isn't an acronym but rather was simply selected as an inspirational name.

57 At http://acs.ist.psu.edu/projects/saoar-faq/soar-faq.html in October 2005.

58 Rodney A. Brooks and Anita M. Flynn, (1989), "Fast, cheap and out of control: A robot invasion of the solar system", in *Journal of The British Interplanetary Society,* 42, 478-485, (1989); Rodney A. Brooks, (1990), "Elephants don't play chess" in *Robotics and Autonomous Systems* 6 (1990) 3-15; Rodney A. Brooks, (1991), "Intelligence without representation" in *Artificial Intelligence* 47 (1991), pages 139–159. Submitted 1987.

This widespread failure to separate text on exposed surfaces from electrons inside storage, has caused enormous misunderstanding of the computer, and now, therefore, also of ChatGPT.

4.5.1 Early avoidance

Many exaggerations and false claims were made in the heyday of symbolic AI in the several decades following 1950. Most AI authors attributed semantic characteristics to symbol systems, but the system lacked these properties.

For example, MIT professor Marvin Minsky's 1968 book, *Semantic Information Processing*, soon became iconic. It contains commentary on current research at MIT, then a hub of symbolic AI research, and a number of computer programs.

But Dreyfus and Dreyfus note:

> "None of the programs in [Minsky's] Semantic Information Processing ...have any semantics, that is, any understanding of what their symbols mean."[59]

Minsky obliquely indicated this, saying:

> "...let us take a look at the programs in this book ... one cannot help being astonished at how far they did get with their feeble semantic endowment."[60]

Translation: zero semantic endowment. Minsky's former doctoral student Joseph Weizenbaum later remarked:

> "I've known [and worked with] Marvin for a very long time ... He loves to say sensational things ... and early on I began to form a hypothesis ... that when he dies, we'll find a letter addressed to us, to be opened after his death. And the letter will say 'Dear children, how could you have believed all the bullshit that I have told you through all these many years?'"[61]

A key symbolic AI device was redefinition of mental terms. Names denoting aspects of human mentality, such as "knowledge", "reasoning", "semantics", "meaning", "understanding", "perception", were redefined to mean elements of symbolic AI theory and practice. A good source of symbolic AI redefinition is Stuart Russell and Peter Norvig's leading AI textbook, *Artificial Intelligence: A Modern Approach*.

Minsky, in his, *Semantic Information Processing*, celebrates this approach of redefinition:

> "Some readers may be disturbed by my deliberate use of psychological terms, such as "meaning," not usually employed so freely in describing the behaviour of machines ... But it is my opinion that these mentalist terms are not all superficial analogies. Indeed, the computer programs described here themselves confirm the validity and fertility of the intellectual revolution that came with the discovery that at least some mentalist descriptions of thought processes can be turned into specifications for the design of machines or, what is the same thing, the design of programs."[62]

---

59 Hubert L. Dreyfus and Stuart E. Dreyfus, (1989), *Mind over Matter*, Free Press, page 70.
60 Marvin Minsky, (1968), *Semantic Information Processing*, MIT Press, page 26.
61 Joseph Weizenbaum, quoted by Daniel Crevier in Daniel Crevier, (1993), *AI: The Tumultuous History of the Search for Artificial Intelligence*, page 83.
62 Marvin Minsky, (1968), *Semantic Information Processing*, MIT Press, page 2.

Well, that's sure a steaming spurt of projectile semi-solid male bovine excreta. Redefinition does a great disservice to research. Such dissembling legitimizes fake science in the guise of great intellectual progress: of "*intellectual revolution*". What Minsky and others did in making their redefinitions was to cripple AI research for over half a century. All for the sake of fame and funds. The same is happening now with LLMs including ChatGPT.

4.5.2 Current avoidance

Today, the terms "knowledge", "understanding", "reading", "seeing", "learning", "perceiving", "hallucinating", "knowing", are redefined to mean elements of connectionist LLM theory and practice.

Semantics is still being avoided. The media frenzy over ChatGPT seems a repeat of excessive claims made in early symbolic AI days. Many LLM fans use similar hyperbole. This spreads misinformation just like Minsky and others over half a century ago.

Billions were squandered on symbolic AI. Billions are likely now being squandered on LLMs. But speculative investors are happy. A careful analysis shows that some LLM experts know what they are saying is false.

For LLM redefinitions, one can listen to virtually any of the many YouTube videos of experts explaining and promoting LLMs, including experts on or recently returned from global pan-continental trans-national planetary nation-hopping promotional world tours. And...

> "OpenAI is in talks to complete a deal which would value the company at $80 billion dollars or more, nearly triple its valuation less than six months ago..."[63]

Yet how bad is AI snake oil vending generally? It seems worse than alchemical theory. At least the supposed elemental substances which adepts called "earth", "water" and "air" had some relevance to today's science. We now know there are no such elements. But the alchemical concepts *earth*, *water* and *air* were much like today's concepts *sold*, *liquid* and *gas*. The adepts mistook *states* of matter for *types* of matter.

Current AI doesn't seem to have that sort of relevance to science. Yet the science of chemistry developed from alchemical equipment and new concepts. The science of Machine Intelligence may develop from AI equipment, the equipment referred to with the shape "computer", and from new concepts.

Both symbolic and connectionist AI use mental terms to refer to non-mental entities. In 1995, philosopher Daniel Dennett, lamented:

> "...the real difficulty arises from the fact that although [AI researchers] are trained as computer scientists they use a lot of terms that philosophers use [when describing the mind], and it takes a long time to discover that they don't mean the same things by them. Their terms are 'false friends'."[64]

In 2012, British physicist David Deutsch made a brutal yet accurate assessment:

---

63 *The New York Times*, 20 October 2023, "OpenAI in Talks for Deal That Would Value Company at $80 Billion".

64 Daniel Dennett, (1995), "In Defense of AI", in Peter Baumgartner and Sabine Payr (eds), (1995), *Speaking Minds*, page 61.

> "...today in 2012, no one is any better at programming an AGI than Turing himself would have been ... The lack of progress in AGI is due to a severe log jam of misconceptions. I cannot think of any other significant field of knowledge where the prevailing wisdom, not only in society at large but among experts, is so beset with entrenched, overlapping, fundamental errors..."[65]

ChatGPT might be seen as "putting the lie" to this assessment. But as per arguments above, it doesn't. ChatGPT in its present form doesn't understand anything, and claims made about it by experts and others are false.

While redefinition makes it easier to explain the technologies, it almost always destroys scientific truth. Most redefinitions by LLM pundits imply that LLM systems have semantic properties and abilities they don't have. And semantics is the essence of intelligence.

Overcoming present hype seems crucial. Without genuine human-like machine intelligence there seems little chance to significantly mitigate climate change or the poverty currently fomenting wars. There's little evidence humans are going to do enough.

Forget the possibility of the robot apocalypse. We need to be realistic. We're faced with the reality of climate Armageddon and with the human military apocalypse happening right now in 2023[66] with the slaughter of tens of thousands of civilians. We need genuinely intelligent machines to save us from ourselves. As climate disasters worsen, as crops fail, as droughts worsen, the human apocalypse will get worse.

Billionaires want AI research paused or stopped, citing annihilation of humanity by the intelligent machine. But who would run the corporations in the event of the intelligent machine? One suspects that the great enthusiasm for AI regulation now gripping western powers has more to do with cementing control of populations by corporations and their political supporters than with the common good.

Long before the intelligent machine has the feral knowledge and skill set to survive on the battle field, intelligent machines could greatly help carbon capture, plastic removal from oceans, flood mitigation, bushfire control, desalination and much more. As if other entities with human-like intelligence should evoke fear and loathing in humans! The greatest threat to humans is humans.

So it seems very important to abandon mythology and dissembling about LLMs, take the LLM elements which are genuine advances, bin the rest, and urgently make progress towards the genuinely intelligent semiconductor substrate.

4.5.3 ChatGPT is partly an advance

I want to argue that ChatGPT ***is*** progress towards genuine human-like machine intelligence. It contains elements which could be used semantically. However, they aren't being used semantically at present.

---

65 David Deutsch, (2012), “Philosophy will be the key that unlocks artificial intelligence”, *The Guardian*, 3 October 2012.

66 As reported by, for example, Lucian V. Truscott IV, (27 October 2023), "The world is coming apart at the seems", *Salon*, salon.com, 27 October 2023.

The main semantic element of knowledge is the relational connective. Structurally, ChatGPT is a connectionist network. Connectionist networks are made of connections and nodes where connections meet. But under connectionist theory, the connections and nodes aren't used semantically.

Rather, the structure, the relationships between nodes, is predefined by a human, and once defined doesn't change. Only values of properties, such as connection weights, change. But as everyone knows, humans learn by the *addition* of connections in real time as a product of sense perception.

One might note that there are no connective elements in Searle's picture of computation, the Chinese room. Baskets of spare text characters are present, certainly. But no balls of string. Yet as everyone knows, the relational connective is fundamental to human knowledge and intelligence. Organic brains are full of connections and nodes. But Searle's picture of computation, based on the Turing machine, has none.

While Searle claims that the Chinese room is a picture of the semantics of the electronic computer, it's really a picture of the extrinsic semantics of the Turing machine. The Chinese room is no help in understanding the semantics of the computer. It still has to be shown, then, that if the relational connective is the key to intelligence, how the computer realizes the relational connective. This is discussed in sections 4.6, 5 and 6.

4.5.4 Turing's avoidance of semantics

If I can make an historical observation, AI's problems concerning semantics started with Turing. He camouflaged the problems and removed semantics from the field of research. As a result, AI has never seriously sought to understand semantics. Yet semantics – understanding, meaning, knowledge – are the essence of human intelligence. AI has never understood semantics. AI has never understood intelligence. The commentary on ChatGPT shows this. Precisely how and why has this gross failure happened?

The Turing test. Founder Turing's 1950 paper, "Computing Machinery and Intelligence", is considered the manifesto and founding document of AI research, and centers on a test for computer intelligence now known as the Turing test. This test is defined as a test of the contestants' observable behavior.

I think Turing conceals his lack of understanding of semantics in three other ways besides predicating his 1950 paper on testing behavior. These all relate to sense perception, the means by which we gain most or all knowledge. Turing had a computational view of intelligence. In my view, he probably couldn't explain sense perception as a computation.

We can say that elements of knowledge are created during the process of sensory transduction. But computation presupposes existing knowledge. It likely can't explain the coming into existence of knowledge. These are the three ways in which I think Turing avoided the semantics of sense perception:

(a) strongly promoting telepathy, a form of non-sensory perception,
(b) recommending only one teleprinter for text communication, and
(c) saying computers internally manipulate what to humans is external: text.

(a) Turing's strong promotion of telepathy

Turing devotes a section of his (1950) paper to E.S.P., extra-, or non-sensory, perception, on (p. 453) saying: "*...the statistical evidence, at least for telepathy, is overwhelming*." Telepathy is a form of perception which avoids the use of sensory apparatus. It's more like water through a hosepipe. This avoids the computational hiatus of sensory transduction.

This hiatus can be indicated as follows. Computation is reaction to values of an inherent property of substance, but the environmental substance which impacts the outward-facing sensory surface and to which the surface reacts does not pass through the sensor. Hence, neither do its inherent properties. The continuation of instances of values of a property is denied by the process of transduction.

In Turing's paper about human-like intelligence, the crucial word "perception" is present, but without the need to discuss or even mention the semantic process of acquisition of knowledge by virtue of the senses.

(b) Teleprinter communication

In Turing's (1950) paper describing his behavioral Turing test for computer intelligence, a judge occupies one room and the two contestants, another. The judge asks the contestants questions and they answer. The contestants are hidden from the judge. Turing (p. 434) recommends: "*The ideal arrangement is to have a teleprinter communicating between the two rooms*".

One contestant is human, the other a computer. Then Turing recommends a type of computer contestant: a 1950s machine, and (p. 442) "*modifying this computer to have an adequate storage, suitably increasing its speed of action, and providing it with an appropriate programme*".

If the judge, after various questions and answers, cannot reliably distinguish human answers from machine answers, then the computer is reasonably said to have a human-like intelligence.

This is a test *of* behavior and *for* the existence of the contestants' *understanding* of text. So fundamentally, the Turing test is a semantic test. The evidence is behavior – pressing teleprinter keys. But what the evidence is *for* is the existence of a semantic process, that of understanding the meaning of the shapes of the judge's text questions. (A process which, when it occurs in humans, is typically called "thinking".)

Thus, although Turing denies the possibility of a machine thinking, saying the idea is "*too meaningless to deserve discussion*", he bases his test for machine intelligence on a process which, when it occurs in humans, is called "thinking".

The judge types a question on the keys of the teleprinter in the judge's room. The contestants see the resulting text print out on the paper in back of the teleprinter in the contestants' room. Then the contestants answer this question by typing their answers on the teleprinter in the contestants' room (anonymously identifying themselves as "A" for computer, and "B" for human).

The contestants' answers then print out on the paper in back of the first teleprinter, the one in the judge's room. The judge sees the text answers and understands the meanings of their shapes (it's assumed everyone knows the language).

On the basis of this understanding, the judge then decides which contestant best understood the meanings of the shapes of the judge's text question.

Thus, the judge's task is to assess the *semantic* process of each contestant (that of understanding the meaning of the shapes of the judge's text questions). And this, when the contestant was looking at the shapes of the judge's text question. Did the semantic process called "thinking" exist inside the contestant? If so, what was its quality?

Now there is something very drastically wrong with this picture as Turing describes it.

So wrong that the judge cannot possibly identify the intelligence (or not) of the machine contestant. As far as I know, in all actual cases of performing the test, such as those in the annual Loebner Prize Competition first held in 1991[67], the setup in relevant respects has been that as described by Turing (1950).

The severe problem with the Turing test as Turing describes it is that it tests only *one* of the contestants. Also, it doesn't test contestant behavior. All it tests is the behavior of the judge's teleprinter.

To understanding the judge's questions, a contestant needs to see the text shapes print out on the paper in back of the teleprinter in the contestants' room. Only one contestant has eyes. Only one can see the questions. Hence only one is in a position to understand the questions then, given fingers to press the keys, to answer them.

The human contestant has eyes and fingers, sees the judge's questions, understands the meanings of the shapes, then types an answer. But the computer contestant is a 1950s machine with improved speed, storage and program. It has neither eyes nor fingers. It can't use the second teleprinter.

So how does the computer communicate with the judge? Answer: It's wired directly into the judge's teleprinter (the annual Loebner Prize Competition, now defunct, used wire or wire substitute such as WiFi or the internet).

What flows between judge and computer is electricity, electrons and their field effects, not text. All the computer is exposed to is electricity. Just like ChatGPT. The computer contestant is never in a position to understand the text questions, or type a text reply. The computer might even be intelligent, but the judge could never know.

This is the same fundamental mistake made when commentary considering ChatGPT. ChatGPT is an electronic entity in a data center and never sees online users' text questions. The Turing test's computer contestant is an electronic entity in the contestant's room and never sees the judge's text questions.

---

[67] See for example, R. Epstein, (1992), "The quest for the thinking computer" in *AI Magazine*, 13(2), pages 81-95; M. Mauldin, (1994), "Chatterbots, tinymuds, and the Turing test: Entering the Loebner Prize Competition". *Proceedings Twelfth National Conference on Artificial Intelligence (AAAI-94)*, pages 16-21; C. Platt, (1995), "What's it mean to be human, anyway?", *Wired*, 1 April 1995; and Hugh Loebner, (2008), "How to hold a Turing test contest", in R. Epstein, G. Roberts, and G. Beber (eds.), *Parsing the Turing test: Philosophical and methodological issues in the quest for the thinking computer,* New York: Springer, pages 173-180.

Did Turing see this problem? I think so. But he tried (very successfully) to obscure it. He says (1950, p. 434): "*The ideal arrangement is to have a teleprinter communicating between the two rooms*". But just a *single* teleprinter, "***a** teleprinter*"? *Two* are needed. What not mention the second teleprinter, the one in the contestants' room, which the computer contestant can't use because it lacks the needed sensorimotor apparatus?

Turing could have simply said that the computer contestant needs to be appropriately robotic. Why didn't he? I think it likely comes back to his inability to explain sense perception as a computational process.

Again, now with his description of the Turing test, as with telepathy, Turing fudges the matter of semantics to avoid attracting attention to the question of sense perception.

(c) description of the computer as manipulating text

The third piece of evidence in support of the proposition that Turing didn't understand semantics but concealed this, concerns his symbol-processing explanation of the electronic computer.

Turing explains the electronic computer as manipulating inner text, mainly numerals (1950, p. 437, quoted below). However, he clearly knew this explanation was false. In (1945) he had designed an electronic computer, the Automatic Computing Engine, or ACE. His 1946 report on the ACE[68] shows that he knew no text is manipulated inside computer electronics.

In 1950 while he was writing his (1950) paper, he was working at the University of Manchester on the Manchester Mark I electronic computer, and writing the programmers' handbook for the soon-to-arrive Mark II (also called the Ferranti Mark I)[69]. A directly-connected teleprinter was used for human interaction with the Mark I machine.

In his (1951) *Programmers' Handbook for the Manchester Electronic Computer Mark II*, (p. 89 – 90), Turing explains the data format of the teleprinter. This was called the "5-unit code", or "5-element code", developed by Gauss and Weber in 1832 and also called the "permutation code". This code was the forerunner of the today's 8-unit binary code called the "byte". The identical 5-unit binary format was used internally in the two Manchester computers. Turing (1951, p. 89-90) about the teleprinter:

> "It may be as well however to explain the operation of a teleprinter, or rather the nature of the signals which are transmitted down teleprinter lines. Ideally these signals change instantaneously from one voltage called mark to another called space. The mark signal also does duty as a 'stop' signal, and the space as a 'start'."

This change from a voltage called "mark" to a voltage called "space" was the binary difference of the 5-unit code. The code comprised five units of either a mark or a space.

---

[68] See for example, B. E. Carpenter & R. W. Doran (eds), (1986), *A. M. Turing's ACE Report of 1946 and other papers*. Cambridge, MA: MIT Press. The PDF is available online.

[69] Alan Turing, (1951), *Programmers' handbook for Manchester Electronic Computer Mark II*, Computing Machine Laboratory, University of Manchester. Photocopy courtesy B. Jack Copeland. The handbook is undated but dated by Copeland (in Epstein et al, 2008, p. 138) as 1950, and includes an errata sheet dated 13 March 1951. According to Turing's biographer Andrew Hodges (1992 [1983], p. 399) Turing was working on the handbook during much of 1950.

The point is that the terms "mark" and "space" for the teleprinter were used to refer to exactly the same things which were named "1" and "0" in electronic computers in Turing's day and still today – either of two possible voltage levels. Turing knew that electrons are not text. Turing knew that no text is manipulated inside electronic computers. However, when describing the computer (1950, p. 437) he says:

> "The executive unit is the part which carries out the various individual operations involved in a calculation. What these individual operations are will vary from machine to machine. Usually fairly lengthy operations can be done such as ' Multiply 3540675445 by 7076345687 ' but in some machines only very simple ones such as 'Write down 0 ' are possible."

Numerals being text. Then he says that the computer store (today called "memory") contains what a human's notebook contains, that is, text:

> "We have mentioned that the 'book of rules' supplied to the [human] computer is replaced in the machine by a part of the store. It is then called the 'table of instructions '. It is the duty of the control to see that these instructions are obeyed correctly and in the right order."

So the store contains rules, and rules are text. Tables of instructions are text. Unless Turing is using the expression in a new sense of non-text. But if so, he doesn't reveal this.

How does saying computers internally manipulate text, when this is untrue, help conceal the problem of the semantics of sense perception?

We could investigate this, but this is not strictly necessary. People, on reading Turing's 1950 paper, easily accept that the computer contestant in the Turing test might understand the judge's text question. I've not seen a semantic objection to the Turing test, though a basic semantic analysis of sense perception in the contestants' room shows that the computer contestant could never be tested.

4.5.5 The semantics of AI's two models

The two overarching AI models are the symbolic theory and the connectionist theory of intelligence, both of which AI seeks to realize inside the electronic components of the electronic substrate, which substrate receives and manipulates electrons and their electromagnetic field effects.

I argued above that AI doesn't understand semantics. It didn't understand it at the start. It hasn't understood it between Turing's day and now, and it still doesn't understand it now. AI has never understood semantics. Semantics (understanding, meaning, knowledge) is the essence of intelligence. AI has never understood intelligence.

I think the Turing test was the first admission of this. A test predicated on the claim that the idea of computer intelligence is meaningless. Thinking is the paradigmatic process of intelligence, and on Turing's account, the idea of a thinking machine is "*too meaningless to deserve discussion*" (1950, p. 442).

AI's explanation of ChatGPT, a chatbot existing in electronic substrate, as trained on text scraped from the internet (which is made of electronic substrate) is merely the latest example of AI's deep lack of understanding of intelligence.

Another admission of AI's lack of comprehension of intelligence is its continual use of human knowledge to define elements of LLMs. In symbolic AI, human intelligence defines the behavior of the machine, mainly through conditionals, if-then's, in the program. The programmer "says", if the input is so-and-so then the output is such-and-such. The programmers knowledge of the world determines which such-and-such goes with which so-and-so. For example, the program when written down: If the input = "How are you?" then the output = "Very well, thank you".

LLMs, by contrast, uses human intelligence to label training sets. There is unsupervised learning but this has proven inadequate on its own. LLM models rely on human intelligence in the "alignment" phase of making "output" conform to human expectations, and in the "attention" phase of adding "context" to the ANN.

It's not that human configuration of the computer must be avoided. Far from it. In a genuinely intelligent computer systems, a humans write algorithms which, in a sense, replicate the results of evolution in the organic case. Turing, (1950, p. 456), "*The experimenter, by the exercise of intelligence, should be able to speed [evolution] up"*.

But human intelligence isn't used to defining output given input. In the intelligent machine, that comes from the environment itself via sensory transduction. The environment determines what is connected to what in the inner semantic structure. Semantic content is extracted from sensory streams. Knowledge comes from the environment via the knowledge-creating sensory process of transduction.

In LLMs, humans decide what is connected to what in the ANN structure. In the cases of "attention", "alignment" and "supervised training", the relevant semantic content is in the brain of the human observer. But in the intelligent machine, it is inside the machine itself, and the yet-to-be-assembled elements of knowledge are inside the streams of data received from sensory apparatus in reaction to the proximate environment.

The LLM process of "token creation", or "tokenization", part of the process of "vector embedding" in "vector databases" and elsewhere, usually doesn't rely on human knowledge to determine the tokens. But these tokens don't come from sensory data streams.

I conducted research on extracting tokens from data streams[70]. When the parameters of the algorithms were tuned down, the algorithms produced tokens very similar to those of LLM token generation. The algorithms didn't define the resulting data, but rather extracted regularity, in data compression lingo, "redundancy", from incoming streams.

What was regular was a feature of the stream itself. What was extracted was determined by the contents of the stream, not by the contents of the algorithm, except insofar as the algorithm set the repetition threshold. The algorithms added connections to the data structures if relevant connections didn't already exist.

---

[70] Respective patents are US 6,414,610 and 5,748,955. The structures and algorithms of 5,748,955 and 6,414,610 are described in data compression terms, but the 5,748,955 provisional application was an AI invention of a "cognitive database". At the time (1993) I was unsure whether an AI patent would be accepted as meeting the practical usefulness criterion of examination.

4.6 Temporal theory of intelligence

Released in late 2022, OpenAI's online chatbot ChatGPT amazed tech savants with its apparent ability to understand text questions online users type. Experts and media alike reported that the chatbot is trained on text found on ("scraped from") the internet. Many said that from this training the chatbot learns the meanings of the text, and that this is why it understands, partly or fully, the text questions online users type.

As we have seen, all these claims are false. The semantics of the computer reveals a very different picture. ChatGPT in its present form doesn't understand anything. And it's not trained on text. It doesn't even receive online users' text questions.

I'd like to suggest a semantic theory of intelligence which explains knowledge acquisition with concepts which can also be used to understand how to program a computer to facilitate the acquisition.

By "computer", I mean the electronic device picked out with the shape "computer", but without implying any type of inner processes. That is, where the shape picks out a certain type of hardware, without indicating what happens inside the hardware.

So the device is called a "computer" in the same sense that the American two-party political system, based as it is largely on bribery, deception, and obeying the will of the donor class, refers to itself with shape "democracy".

The semantic theory starts by asserting that semantics is the essence of intelligence. And further, that the central semantic process is the inner process called "thinking".

This seems to reflect quite closely the popular notion of intelligence. It would be generally agreed that thinking is a semantic process, that it fundamentally concerns understanding and knowledge, and that thinking is the essential process of intelligence. If this is correct, then it seems both remarkable and an indictment that semantics has never been at the forefront of AI research.

Turing ended his foundational (1950) paper "*We can only see a short distance ahead...*". Putting semantics at the forefront of research could be the next step ahead, one which moves on from Turing's abandonment of the essence of intelligence.

On this view, it's time to put to rest founder Turing's claim that the idea of a thinking machine is "*too meaningless to deserve discussion*". In remains to be seen, however, what thinking amounts to expressed in concepts of the semantic theory.

Terminology. Semantics (knowledge, meaning, understanding, thinking, intentionality) is the essence of human intelligence. It seems useful at this point to briefly explain these items in general terms. Knowledge is a semantic *object* comprising semantic *structures*. A meaning is a semantic structure. Intentionality is a semantic *property* of semantic structures related together. Understanding is a semantic *processes* operating within and between meanings.

Method. Little is known about the mind in terms applicable to also understanding the computer. Thus, it seems very difficult to imagine how a computer could be configured to have or acquire a mind. In order to build a car engine one first needs to know how the internals of the intended type of engine work. But we don't now how the internals of the mind work in terms which explain how to build one. Most leading researchers including Geoffrey Hinton agree that the connectionist process of "learning", back propagation, is not a neural process.

The solution is to chose a place to start trying to understand the mind, which place we do know quite a lot about in what seems the needed type of technical detail.

But which place is that? It's the place where new concepts can most likely be developed safely. That is, developed with reasonable certainty that the concepts will be appropriate to the purpose.

We also want the place to be the coalface, the place where the elements of intelligence are created. This means that concepts developed downstream from the coalface will have a better chance of also being appropriate. We want to start at the place that doesn't rely on other newly developed concepts.

So what is this place?

We are talking about semantics, principally understanding and knowledge. But understanding of what? Knowledge of what? Knowledge of the outer. And understand *by* what? Understanding by the inner. The coalface is the place where the inner reacts to the outer. This is where knowledge, or at least the components of knowledge as yet uncombined, are created.

Thus, the starting place in seeking to understand intelligence with appropriate concepts is sensory apparatus. Effectors can be set aside for the moment.

Need of a principle. The next step after deciding on a method and a place to start is to state a semantic principle of knowledge acquisition by way of sensory apparatus.

To understand how a computer could acquire knowledge in the same sense we do, we need to understand the role human-like sensors play in knowledge acquisition.

The sensor is the interface between the inner and the outer, the mind and the environment. Sensors react to the environment. This reaction includes sending streams of data into the inner world to the central system. These streams must contain knowledge, either holistically or as uncombined elements, in some form. So what is this form?

This is a difficult question to answer. We know that environmental substance doesn't survive the sensory transduction. The impinging molecules, atoms, photons don't pass thought the sensor and into the inner word. The sensor is a substance barrier, not a substance portal. But something *must* pass thought. How else could the inner come to know the outer?

For a solution to this dilemma we look to the fundamental ontology of that which exists: *substance, property, relation* and *time*[71]. We know that substance doesn't survive transduction. Since property is inherent to substance, neither does property. It follows that what does survive must be either *relation* or *time* – or both.

[71] See for example, Wikipedia under the search term "ontology", and *The Cambridge of Philosophy*, Cambridge University Press.

The semantic theory says both, since we have knowledge of both relation and time. But on the other hand, substance and property don't survive, but we do have knowledge of substance and property. So how do we have knowledge of external substance and property? This needs to be explained, and seems to relate to intentionality, or aboutness.

To this analysis just above we can add that sensory streams must also contain something which gives us knowledge of *causation*. This must be a very basic component of sensory data, since many types of animal depend on knowledge of causation, in some sense of "knowledge", to survive.

I'll indicate, below, that this further stream element is *repetition*. A system can use repetition thresholds to distinguish likely accidental conjunction from likely causal conjunction.

Space, time and causality. Gary Marcus and Ernest Davies recently (2019) argued that "*Computer systems need to understand time, space and causality. Right now they don't*"[72]. And further:

> "The problem is not that today's A.I. needs to get better at what it does. The problem is that today's A.I. needs to try to do something completely different".

I agree with Marcus and Ernst that AI needs to do something different. But not *completely* different. Connectionism already has elements which could be semantic, but uses them in non-semantic ways. This is discussed in detail section 6.

Time and space can be reduced to the fundamental ontology. Space can be reduced to *substance, property,* and *relation.* And time is *time.* I'd now like to consider these four ontological types, *substance, property, relation* and *time*, from the perspective of sensory transduction, the place where the elements of knowledge are created.

Substance and property. Sensory streams comprise units. In the computer case, these are made of clocked electrical current. One unit is a clock "tick" or cycle". A such unit is a quantity of current of a certain voltage which doesn't change (small fluctuations are ignored). This current is really continuous, but the hardware treats it as divided into discrete units.

A stream by definition moves relative to a point, line or surface. As a stream of units passes a point, over aline or through a surface, it does so one unit after another in time. For example, at a given surface, one unit arrives and can be reacted to, then another unit arrives and can be reacted to.

We can now consider the types of possible reaction. One type of possible reaction to a unit arriving at a surface is to the *substance* only. For example, in the Chinese room, discrete Chinese ideograms drop from the slot in the door. The rulebook's instruction might be to place the ideograms alternately to the left and to the right of the slot. The shape of the ideogram is irrelevant. The instruction concerns just substance. Its properties are irrelevant.

---

[72] Gary Marcus and Ernest Davies, (2019), "How to build artificial intelligence we can trust", *New York Times*, 6 September 2019. Marcus has authored a number of opinion pieces for the New York Times and other publications on the failures of the deep learning statistical modeling of AI.

Another type of possible reaction to a unit is to *values* of a *property* of substance. This type of reaction is in fact what the rule book mandates, since this is the type of reaction performed in computation, the type when "*writing certain symbols on paper*" (Turing, 1936, p. 249).

Taking these two types of reaction, we can see that a reaction to an incoming sensory stream unit can be either to *substance* or to *property.*

Relation and time. Aristotle taught the semantic relevance of time:

> "...as to the question of which of the faculties within us memory is a function, (it has been shown) that it is a function of the primary faculty of sense-perception, i.e., of that faculty whereby we perceive time."[73]

Suppose that the Chinese ideograms which drop from the slot in the door into the Chinese room are made of cast iron. Each pair of ideograms could be tied together with a length of string. The shapes of the metal ideograms are irrelevant. Whatever the shape, one it connected to the next by string.

Also, properties of the string are irrelevant, its length, mass, color and so on are irrelevant. All that matters, to record the two ideograms' adjacency in time as they enter the room, is that the pair are connected. The string permanently records in storage the fleeting relationship as the units enter the room one after the other.

That is, in the stream there is a *relationship* in *time*. This is recorded as a relationship in substance (the string). Thus in a pair of units passing through a surface, all four fundamental ontological types, s*ubstance, property, relation* and *time* are present. *Substance* and *property* in each of the units of the pair, and *relation* and *time* in the pair as a pair. These four types can be recorded by storing the units (substance and property) and linking the units together with other substance, connective elements.

The relation of temporal contiguity *per se* is that without which there would be no relationship. It might be argued that this idea of temporal contiguity *per se*, is really an occult entity and really there is no such thing. In imagination, this idea of disembodied contiguity seems vaporous, ethereal and hard (or even impossible) to believe.

But I think the temporal relationship of togetherness in time, *per se*, clearly is a"thing" in the most important sense. It can be recorded as a physical connection. Then that stored record can be turned back into a new instance of the temporal stream.

Causation. We can say that there are two sorts of causality. In one, an external object produces (or reflects) particles which impinge on a sensor's detecting surface. The sensor reacts to the particles but not to the object which causes or reflects them (except in the case of touch). Part of the sensor's reaction is to create units and emit them in a stream.

There's a 1:1 correspondence between the properties of an impinging group of external particles and the properties of the internal units. The internal units don't have the same properties as the particles. But when a different group of the same type of particles arrives

---

73 Aristotle, (c. 350 BCE), "On Memory and Reminiscence", translated by J. I. Beare, available online at eBooks@Adelaide, University of Adelaide, Adelaide, Australia, retrieved in 2004.

with the same value of the same external property as before, the respective inner units are given the same value of the same property as before.

Knowledge of causation must have a very simple foundation. This is repetition. The central system receives sensory streams then applies a repetition threshold. In a given stream, pairs of units of values repeated over a threshold are stored as structure. Given that the threshold is adequate, this stores likely causation. Repetition below the threshold is discarded as accidental conjunction.

In the first sort of causation, the repetition is within a given sensory stream. The source of the impinging external particles is an instance of a type of external object, for example a bell. The groups of arriving particles will have values which will be repeated over time. This will result in an internal structure the source of which was one or more bells.

Such structures are unique to a given sensory modality, or perhaps even unique to a given sensor, for example in the case of binocular vision. Once created, such a structure can be activated when an instance of the type of source is present again. When structures of different modalities are repeatedly activated close together in time over a repetition threshold, the inner structures are connected. The resulting multi-modal structure is called an "object structure", meaning type of external object. The bell object structure, for example, will likely comprise a structure derived from vision sensors connected to a structure derived from sound sensors.

The second sort of causation is when object-type structures already exist. For example, when the structure caused by *fire* and the structure caused by *smoke* already exist. These two inner structures are activated. A repetition threshold of the pair of activations is applied. If the two activations are repeated over the threshold in a time period (perhaps a day) then the two inner object structures are connected.

Timing of multiple streams. There could be two or more streams of sensory data arriving at a surface, for example from different sensors. Just considering two streams, one would say that a unit in one stream might arrive at the same time as a unit in the other stream. But close examination would likely reveal that one unit arrives after the other. However, to a system based on clock cycles, if two units arrive at different locations within the same clock cycle then, to the system, the units arrive at the same time.

The structural connective. In the picture described in this section, instances of the relation of temporal contiguity in a stream are recorded as connectives in semiconductor storage. A connection records the temporal relation *per se*, that without which there would be no relation.

In the experiments I did, the structures were "trees" in a "forest". The forest had different areas, one per sensor. Within a given area there was only one set of leaf values (leaf nodes). All the trees in that area used leaves from this single set. Hence there were often many outgoing connections from a single leaf. The forest was a vast record of contiguity in time, the fundamental element of knowledge.

Temporal contiguity survives transduction. The only thing that survives transduction is togetherness in time. It survives as a duplicate. It's the same thing on the inside between emitted units as just happened on the outside between impinging groups of particles.

That's what allows the inner to come to know the outer. Transduction creates the semantics of knowledge. This is to be expected. Transduction is the principal process of the sensor, and the sensor is the place where the outer affects the inner.

The relationship of togetherness in time combined with the different values[74] of the properties of inner units of streams, and repetition count, is what knowledge is made of.

This, just above, is the basic idea of the semantics of knowledge. It's a possible starting point for new ideas which might be used to program a computer to facilitate the acquisition of knowledge including knowledge of the meanings of words. In this new explanation, the inner connective element, an element of structure, records adjacency or sequence in time in sensory streams, and in the environment impinging on the sensor's detecting surface.

Realizing the connective element in semiconductors. Computer storage is called "linear". Of course it's really 3-dimensional. "Linear" means that it's accessed via a linear (1-dimensional) sequence of identifiers, typically regarded as numerals starting at the shape "0" and incrementing by one. So how is the manipulable structural element realized in computer storage? How to create, find and use them is explained and exemplified in Section 6, "Realizing the theory".

Motor apparatus. The newborn in the cot staring at a rotating mobile suspended overhead is seemingly only doing vision. In looking at the rotating shapes of the mobile – only vision is seemingly involved. However, muscles control the eyes. But we start with just vision. Some motor action is automatic. For volitional motor action to achieve a goal, such as grasping a cup which has handles, first there needs to be recognition of the object, the cup. So first there is a process which establishes inner structure by reacting to sensory streams.

5. ASSESSING THE THEORY

The main idea behind the temporal semantic theory is that the atomic elements of semantics, the only "things" which survives sensory transduction, are instances of the relationship of togetherness in time *per se*. Such association in time can be recorded as connective elements of structure.

5.1 Recording adjacency in time

How could a computer realize and manipulate in electronic storage a relational connective? This question seems easy to answer, but the idea is contentious because connectives seem to be fundamentally different from symbols (text characters) and computation manipulates just symbols. Turing (1936, p. 249): "*Computing is normally done by [a human] writing certain symbols on paper*".

---

[74] See for example, G. Spencer Brown, (1972), *Laws of form*, Bantam Books, pages 1-2.

The established wisdom says that computers operate by internally performing computations, manipulating symbols and nothing else. Searle (1997, 9), "*A digital computer ... manipulates symbols and does nothing else*". There are no baskets of connectives in the Chinese room, Searle's picture of computation. The Turing machine, which many say defines machine computation, lacks the operation, *follow*. But to *follow* is fundamental to using a connective element.

Computation treats text characters differently depending on their values (shapes). Turing's second example of his machine (1936, p. 234) reacts differently depending on whether the scanned text character is shaped "0" or "1". But the values of a connective are irrelevant to its use.

Two key question are, then, can computers internally manipulate connective elements (create, delete, move, follow)? And if so, are these elements a genuinely different manipulable type from the symbol, or are they merely closet symbols? I'd now like to consider these two questions.

5.2 Connective elements of ChatGPT

ChatGPT is a large language model. These are artificial neural nets comprising connections and nodes.

We can agree that understanding text presupposes reacting to its shape. For ChatGPT to understand text, then, it needs shape-detecting sensory apparatus. But it has none. It's an electronic entity existing in semiconductor components of semiconductor substrate. The substrate might have sensors for detecting overheating, but none for detecting shape.

ChatGPT, I argue, does have elements which could be used semantically as part of the process of understanding external text the same way we do, by detecting it with shape-sensing apparatus. The code presented in section 6 shows how such elements, relational connectives, can be created, found and used.

Yet there seems a severe problem with realizing connective elements in computer storage. The substrate lacks manipulable physical links. There's no way to create, move, delete or follow physical links. The substrate manipulates electrons. However, the definition of the connective can be realized in other than direct connection with physical substance. The definition doesn't require physical substance stretching from one place to another.

The connection is defined as two ends plus a means of the relevant system to get from one end to the other.

The ends are certainly physical places. But the operation of getting from one to the other need not be that of following a wire. There can be other sorts of "following".

An algorithm "follows" a "virtual connection" by using direct memory addressing and indirection operations. One storage location stores an address, or "pointer". This is an address of another location. An algorithm loads the address into a register then execute a direct addressing operation with indirection. This returns the contents of the second location – the one pointed to.

The algorithm "moves" along the "virtual connection" from the first location to the second location. The contents of the second location might be another pointer. Direct memory addressing and indirection are discussed and illustrated in section 6.

5.3 Are connections fundamental?

One might object that while virtual connections are possible, they are not fundamental to the computer, but rather are compounds of storage place and content. And further, that to understand how a computer could be intelligent one needs to understand the fundamental nature of the machine. And the relational connective is not fundamental.

We can grant the need of understanding fundamentals. But fundamental to what? Intelligence is semantic. The key is semantic content, or knowledge. The manipulable relational connective might not be fundamental to electronics, but it ***is*** fundamental to human knowledge, and we learn by addition of connections. Since at least Aristotle, contiguity has been identified as key to mental processes:

> "...we hunt up the series (of kineseis) having started in thought either from a present intuition or some other, and from something either similar, or contrary, to what we seek, or else from that which is contiguous with it. Such is the empirical ground of the process of recollection."[75]

Connectionist structures, artificial neural nets, are made of connections and places where they meet. Hence ChatGPT contains semantic elements fundamental to knowledge. Or rather, it contains potentially semantic elements. LLMs contain connections but don't use them in semantic ways. The semantic way was discussed earlier. This is to use them to record temporal contiguity which occurs in sensory streams. LLMs don't use connections in this way.

5.4 Are connections closet symbols?

Searle defines "symbol" as a unit of substance whose shape has been assigned linguistic meanings, that is, text. He says computers operate by internally manipulating symbols.

He then expands this concept of *symbol* to include units of substance whose shapes have not been assigned linguistic meanings, but nevertheless are received by a computer from sensory apparatus. One might ask, are connections really just closet symbols?

A connection is defined as two ends plus a means to get from one to the other. This implies a "follow" operation. But in the *follow* operation, an algorithm doesn't manipulate the connection. The motor vehicle doesn't manipulate the freeway. Turing machine manipulations are: create ("print"), identify the shape of ("scan"), destroy ("erase"), and move ("left" or "right").

Following isn't creating, destroying, manipulating or identifying. This indicates that characterizing the computer in terms of manipulation understates its causal capacities. The *follow* operation can determine output but nevertheless isn't manipulation. The idea of manipulation is really a child of the concept of computation.

---

75 Aristotle, “On Memory and Reminiscence”, translated by J. I. Beare, available online at Gutenberg.org.

Computing, in Turing's 1936 sense and Searle's 1980 sense, is reaction to shape. The Turing machine is founded on reaction to shape. The idea of computational manipulation has been used to understand the causal capacities of the computer (so called).

But an algorithm which follows a connection isn't reacting to the shape of the connection or identifying the connection. But rather is seeking to get form one place to another. This is quite different from reacting to shape. All the connections in a system might have exactly the same shape. Whether they do or not is irrelevant to the use and purpose of the connection.

It might be claimed that a connection could be realized in a Turing machine. If so, there are still the two ends, which are locations. That is, the connection is still not a symbol but rather a combination of location and symbol. The Turing machine connection would not be a closet symbol.

The difference between symbol and structure can also be explained this way. Symbols populate structure. Structure itself is of a different ontological type from that which populates structure. Structure is relational, not qualitative. Symbols are qualitative, and a structure may or may not be populated[76].

6. REALIZING THE THEORY

6.1 Realizing a virtual connection

A "virtual connection", can be realized in computer storage using storage location identifiers ("addresses") as pointers, the technique of indirection, and the operation of direct memory addressing.

One location stores the address of another location. These are the two physical ends of the connection. The address is then loaded into a register and a direct memory operation performed with indirection. This returns the content of the second location, the location pointed to, which content might be another address.

Thus, the algorithm starts at one location, that which holds the pointer, then returns the content of second location. The algorithm "follows" the connection from the first location to the second location then as a bonus gets the content of the second location. If this content is another address then the algorithm can progress from one connection to another. This provides a means to quickly travel through semantic structures.

This method of the virtual connection is commonly used in computer applications. As Drew McDermott notes "*Almost everything in an AI program [a semantic net] is a pointer*"[77].

---

[76] In today's electronic computers, when switched on, and because of the nature of semiconductor storage, storage is always populated. The hex value FF is often initialized to all "free" storage locations, and is then construed as meaning "empty", and internal lists are maintained of which storage locations are free.

[77] Drew McDermott, (1976), "Artificial Intelligence Meets Natural Stupidity", in *SIGART Newsletter*, 57, April 1976, and in Haugeland, *Mind Design*, page 160. With semantic nets, the semantics is in the observer of the shapes displayed on screens and printouts, not in the net itself.

6.2 Code for following a connection

Two examples of code for following existing connections are given, the first example in C[78] and the second in assembler[79]. The connection structure is called a "binary data tree" ("tree"). A node is where connections join. "Binary" means that each node has two "children" usually called "left" and "right", apart from leaf nodes which have no children.

The code below, when executing, "walks" the tree starting at the root node then following the connections, or branches, in a certain order. When it reaches a leaf it copies the second field of the leaf record into an output array, as explained step by step, just below. This process, in the terminology used below, is called "unpacking" the root address.

The code is:

```
void unpack_address(unsigned long *cca)
{
if(*cca == 0) {*pap = *(cca+1); pap++;} /* dv found               */
else {unpack_address(*cca);             /* go down one level left  */
      unpack_address(*(cca+1)); }       /* go down one level right */
}
```

The `*` is the indirection operator and can be understood as "what is at"; `cca` is the current connection address, the address of the current connection record, and this is passed to the algorithm by the calling algorithm (not exemplified). The above algorithm starts at the root of the tree. The address of the root is passed to the above algorithm by the calling algorithm.

The first line, (`if(*cca == 0)`) "asks", is the first field of the current connection record NUL (is the record a leaf record)? If so, then `*pap = *(cca+1)`. Copy what is stored in the second field of the leaf record to the current position in the output array. What is in the second field is identified by `*(cca+1)`. The current position in the output array code is identified by `*pap`, the process array pointer.

The content of the second field of the leaf record is called a "data value" or "dv". The value `pap` is the pointer to the current position in the output array, and `*pap` references the current position itself, which is the first vacant location after the end of the output so far. Then, having copied `*(cca+1)` (the content of the second filed) into `*pap` (the first free location in the process array) the algorithm increments the process array pointer by one, `pap++`. This then points to the first free location after the new end of the array.

The next two lines cater for the case where the current connection address is not the address of a leaf record. That is, where there is a left and right child connection. The address of the left child's connection record is stored in the first field of the non-leaf current connection record, and the address of the right child is stored in the second field.

Is this computation? The above code accesses a connection structure realized virtually in computer semiconductor storage, and the code exemplifies how an algorithm can follow the connections.

---

[78] For the C programming language, Brian Kernighan and Dennis Ritchie, (1978), *The C Programming Language*, Prentice Hall.

[79] Intel assembler for the x86 processor.

In walking the tree, the algorithm treats each data value (leaf value) the same way. When found, it is simply copied to the output array. The algorithm doesn't react differently depending on the value of the leaf.

Yet it might be objected that a copy operation is a different reaction depending on the value of the thing being copied. The reaction is to create an object of the same value in a different place.

This objection can be granted. But the point is that the algorithm applies the same operation to all data values, the *Copy* operation. The value of what is copied, the value the second field, isn't part of the algorithm. But in the Chinese room picture of computation, the shape (the value) of the Chinese ideograms ***is*** part of the instructions in Searle's rulebook (the program, the algorithm). Searle as the man in the Chinese room:

> "Now suppose also that I am given a third batch of Chinese symbols together with some instructions, again in English, ... these rules instruct me how to give back certain Chinese symbols with certain sorts of shapes in response to certain sorts of shapes given me in the third batch."[80]

A human copying a list of numerals from one sheet of paper to another would not normally be said to be performing a computation.

In Searle's Chinese room, in his picture of computation, the book (the program) instructs the man (the CPU) to do different things depending on the different shapes of the Chinese text symbols he receives. The Chinese shapes in the rulebook determine the output. But in the structural case exemplified in C above, values stored in the structure determine the output. Not in the algorithm.

When these leaf values come from sensors, the values were created as a reaction to the environment. The program simply copies whatever is the leaf value. This makes for very small and fast programs compared to the combinatorial explosion typical of AI programs which contain rules about the specific shapes of symbols[81].

The above unpacking algorithm does the same thing to each leaf data value. Adds it to the output array. The output array can determine the behavior of the machine. In this case, the human creator of the algorithm doesn't determine the behavior. Rather, it's determined by the content of the structure. And the connection structure is built from what follows next in time after what, over a repetition threshold, in the sensed environment.

It might be argued that a repetition threshold is the same thing as counting by numbers, so computation in the sense of counting by numbers does take place. But a threshold might be just a bucket, and when the bucket is full the threshold has been reached, and there is no counting.

Of course some people say that every process is computation, but that idea goes quite strongly against Turing's 1936 conception. His 1936 conception underpins the common

---

80 John Searle (1980), "Minds, Brains, and Programs", in *Behavioral and Brain Sciences*, 3 (3), 1980, page 419.

81 For combinatorial explosion, see Sir James Lighthill, (1972), "Artificial Intelligence: A General Survey", in J. Lighthill *et al* (eds). *Artificial Intelligence: A Paper Symposium*, (July 1972), The Science Research Council of Great Britain.

understanding of the computer, and on this 1936 conception, computation is the manipulation of text. And this is a very erroneous view of the computer, so called.

In the Turing machine example just below (Turing, 1936, p. 234), the values reacted to are the shapes "0" and "1". The reaction of the algorithm is different depending on the shape. If the shape is "0" then the machine moves its ovipositor two locations to the right then lays a token shaped "1". But if the initial shape is "1" then the deposited egg is appropriately shaped "0".

234 A. M. TURING [Nov. 12,

If (contrary to the description in § 1) we allow the letters L, R to appear more than once in the operations column we can simplify the table considerably.

m-config.	*symbol*	*operations*	*final m-config.*
	None	$P0$	b
b	0	$R, R, P1$	b
	1	$R, R, P0$	b

The idea of reacting differently to different shapes is the main idea of simulation. The different shapes are the different shapes of the description of the system to be simulated. Core to such descriptions is the conditional "if-then" statement. The conditional can be expressed in the form: if the input = "A" then the output = "X" but if the input = "B" then the output= "Y". Typically, a description of a natural system or of a machine contains many conditionals.

Conditional statements used in describing complex systems with many possible inputs quite quickly lead to combinatorial explosion, as per Sir James Lighthill's 1972 report[82] which effectively ended for several decades most government funded AI research in the UK. The Turing machine above executes conditional statements. Each line is a conditional. Given that the Turing machine defines machine computations, this indicates that computation is quite inappropriate as an explanation of intelligence.

Assembly language example. An assembly language version of the above simple tree-walking algorithm more precisely shows the role of direct memory addressing and indirection. The code below comes from work conducted about 30 years ago, and assumes the symbolic paradigm, and hence talks about numbers (numerals) and more generally, symbols, as that which are manipulated.

The routine is called "decompress c/word" (decompress codeword) but it operates on what I originally described as a "cognitive database".

```
;---------------------------------------  in: edi = con#
; decompress c/word                      out: symbol(s)
;---------------------------------------
Decompress proc                           ;
        shl  edi,4                        ;con offset
        cmp  word ptr es:[ebx+edi+0],0    ;is this an I-con?
         je  SymFound                     ;yes
```

82 Sir James Lighthill, (1972), "Artificial Intelligence: A General Survey" in J. Lighthill *et al* (eds.), *Artificial Intelligence: A Paper Symposium*, (July 1972), The Science Research Council of Great Britain.

```
        push  edi                         ;cur con offset
       movzx  edi,word ptr es:[ebx+edi+0]  ;e1#
        call  Decompress                  ;go down lhb (left-hand branch)
         pop  edi                         ;con# nlu (next level up)
       movzx  edi, word ptr es:[ebx+edi+2] ;e2#
        call  Decompress                  ;go down rhb (right-hand branch)
         ret
SymFound:
        call  AddSymbolToBuffer2           ;populate o/put buffer.
         ret
Decompress endp
```

This routine walks a data tree starting at the root node. Data trees, if thought of as 2-dimensional, are like inverted natural trees that have a single root. The data tree "root node" is at the top and the leaves at the bottom. A collection of data trees is called a "forest".

The first line. The routine starts: `shl edi, 4`. The con# (connection number) of the root node has already been loaded into register `edi`. The first line multiplies the content of `edi` by 16 by shifting its bits four positions to the left, then padding the right-most four bit position with nulls (hex zeros). This shifting converts the con# into the offset address of the root node connection record from the start of the forest. The address is the address of the first byte of the connection record.

This offset is the relative address of the root node from the start of the forest. It's the address of the storage location of the root node relative to the first location of the forest. This location is where the root node connection record starts.

The root node connection record (and all other connection records in the forest) comprises eight "fields". Each field comprises two storage locations of eight bits each, making 16 bits in total per field. That is, each field is one "word" long where a "word" is a storage location comprising two bytes, or 16 bits.

Since a connection record is eight fields long, and since each field is two bytes long, the record is 16 bytes long. This is the reason for multiplying the con# by 16 to yield the offset address of the root node from the start of the forest.

An addresses identifies the location of a single byte of storage. The address 0 is the identifier of the first byte of total storage, address 1 is the identifier of the second byte, and so on. An address relative to the start of all storage is called an "absolute address".

The second line. The second line, `cmp word ptr es:[ebx+edi+0],0`. "asks" is the value of the first field of the connection record equal to hex zero? That is, does it contain the semiconductor states named "0000000000000000" (also "null", "NUL", "0x00", "hex 00")?

The line finds this first field by resolving the expression, `es:[ebx+edi+0]`. "es" is the name of a memory segment. In the above code, `es:` holds the absolute address of the start of the segment. Register `ebx` holds the address of the start of the forest relative to the start of the segment. Register `edi` holds the address of the start of the connection record relative to the start of the forest. Adding "`0`" to the absolute address isn't strictly necessary but conforms to the format where the offset of the start of the field within the record is added, and the offset within the record of the first field is zero.

Now it might be supposed that the presence of adding addresses equals the presence of computation. But we need to consider the code from the standpoint of semantics. The "+" shape above has a meaning. When I wrote the code I knew the meaning. But when I typed the above line the only place the "+" occurs is on my keyboard. It didn't leap off and careen into the wire and into the computer.

All that leaves a keyboard is electricity. All that is stored are semiconductor states and electrons, not text characters such as "+". The shape "+" is only computational in that it has a computational meaning. But the groups of electrons and semiconductor states of semiconductor storage are semantically vacant. They have no meanings at all in the sense that text characters do.

In Turing's (1936) sense of, "*Computing is normally done by [a human] writing certain symbols on paper*", semantically speaking no computation occurs inside a computer. The shapes written on sheets of paper have certain meanings to the entity performing the activity. But the computer has no access to the shape "+" which I type, let alone its meaning. So in the sense that humans perform computations on symbols, computers don't internally compute on anything.

The second line in the above, having found the location of the first field of a connection record, then asks, is the content of this location null (hex zero)? Again, of course, in reality the content is not hex anything but rather a conductivity state or collection of electrons. But a convenient textual fiction is used which says a text character is stored, but in reality the text character is merely a name which humans to refer to the electronic states and groups of electrons of DRAM and other forms of semiconductor storage.

The comment to the second line says "`is this an I-con?`". The term "I-con" is short for "interface connection". The idea, here, is that the leaves of the forest are the places where, in the organic case, the various insulated conductors transporting unary pulses from a sensor enter the central system. This system is conceived of a system inside a spherical surface. The leaves are the places on the surface where unary pulses enter the system.

But in the electronic case, the data is not unary but, rather, various values of data from a given sensor usually travel along a single conductor, and the data is demultiplexed before it hits the surface. Each leaf is a different possible data value (or data value received so far) from the electronic sensor.

There are different areas on the surface, one area per sensor. There is only one leaf per data value in a given area. Many trees grow during early sense experience from the leaves in an area, as suggested in US 5,748,955 and 6,414,610.

The third line. The third line, `je SymFound` then "says" what to do if the value of the second field *is* zero. This action is to jump ("`je`", or jump if equal) to the label `SymFound`. After the call to the procedure, `SymFound` , the `decompress c/word` routine then does what is prescribed immediately below the label. This is to call the procedure `AddSymbolToBuffer2`. Buffer2 is the output array of the earlier C code.

Fourth line. The fourth line "tells" the computer what to do if the content of the first word of the connection record is not NUL That is, if it is not a leaf record. The fourth line, `push edi`, saves the offset address of the current connection record from the start of the forest (this offset address is in register `edi`). It saves this address by copying it then "pushing" the copy onto the top of the "stack".

This is being done because register `edi` is now going to be filled with a different address, but the old address will be needed later when the routine "moves" back up the tree structure after encountering a leaf. (The value at the top of the stack, the recently pushed connection number, is moved back into `edi` from the stack by: `pop edi`. This isn't a copy operation but rather the value once copied is then removed from the stack.)

Fifth line. We are now at the line which is the point of this examination of assembly language (a language to the human programmer, text to the human programmer, but when the program is loaded, collections of electrons inside the computer), the line:

```
movzx edi,word ptr es:[ebx+edi+0]
```

This copies the contents of the first field of the current connection record into register `edi`. The previous content of `edi` was the address of another connection record, one end of the virtual connection. The content of `edi` is now the address of the other end of the connection, also a connection record. Hence the routine has "followed" the connection between one storage location and another.

The further assembly code lines complete the walking process.

Again, as with the earlier C code, the assembly language routine adds whatever leaf it finds to an output buffer.

The C and assembler algorithms are very simple algorithms. They merely follow connections, and when a leaf is reached, copy its value to an output buffer. In a sense, this structural form of processing swaps complexity of algorithm for complexity of structure, and the algorithm is very simple.

Whereas in the computational case, structure is very simple or non-existent, and the algorithm is complex, as Sir James Lighthill explained[83]. This non-computational approach of complex structure and simple algorithm seems to offer quite useful survival advantages. Perhaps in the organic case, complex structure is developed incrementally during sleep over months or years, and simple algorithms offer quick response when awake. And quick response when awake is better for survival than no response while asleep.

Simon's famous "ant"[84] thought experiment externalizes complex structure such as can occurs in churned up sand on a beach. And the ant has only a very simple inner process. The complexity is in the proximate environment. Simon was hoping to overcome problems

[83] Sir James Lighthill, (1972), "Artificial Intelligence: A General Survey", in J. Lighthill *et al* (eds), *Artificial Intelligence: A Paper Symposium*, (July 1972), The Science Research Council of Great Britain.

[84] Herbert A. Simon, (1969), *The Sciences of the Artificial*, MIT Press, pages 23 and 52.

such as the computational problem of combinatorial explosion. But human brains are not ant brains. They have great inner structure.

The point is that the process of tree walking is a process of moving from place to place, which is the sort of thing that happens inside organic brains. And it's not computational in Turing's sense of "*writing certain symbols on paper*"(1936, p. 249), or in the sense of what happens in the Chinese room (Searle, 1980), since no reaction to shape is involved in either then organic or semiconductor case.

As I've indicated, I think AI's conception of the computational mind and of the computational electronic digital "computer" is a big problem for AI. Today's understanding of ChatGPT seems deeply infected with the following two myths: the myth of the computational mind and the myth of the computational so-called "computer". The Chinese room argument is perhaps the paradigmatic case of the infection. And Turing's (1950) paper is "patient zero".

7. SEEING FURTHER AHEAD

Turing ended his renowned (1950) paper, "*We can only see a short distance ahead, but we can see plenty there that needs to be done*." It's been over 70 years. Plenty has been done – in pursuit of symbolic and connectionist theories of intelligence.

But, as argued above, neither of these theories explain semantics, the essence of intelligence. A fixation on computation has been counterproductive. The principal matter is the semantics of electronics. How could semiconductors understand in the same sense we do?

The symbolic-computational misunderstanding of the computer has entrenched two fundamental errors in AI theory and practice. First, that computers are essentially Turing machines and hence operate by internally manipulating text. Second, that computers can internally manipulate only one type of thing.

Yet it might be countered that every electronic engineer knows that the first claim is false: semiconductors don't store or manipulate text. But then everyone says ChatGPT is trained on text from the internet, and the internet is made of semiconductors.

There is great confusion which seems like double-think. So much so that David Deutsch, observing from outside of AI theory and practice, felt compelled to conclude:

> "...today in 2012, no one is any better at programming an AGI than Turing himself would have been ... The lack of progress in AGI is due to a severe log jam of misconceptions. I cannot think of any other significant field of knowledge where the prevailing wisdom, not only in society at large but among experts, is so beset with entrenched, overlapping, fundamental errors..."[85]

It might be said that ChatGPT "puts the lie" to this dire assessment, but as I've argued above, it doesn't. ChatGPT has no intelligence at all, and this is not recognized because of much confusion and the myth that computers internally compute.

---

[85] David Deutsch, (2012), “Philosophy will be the key that unlocks artificial intelligence”, *The Guardian*, 3 October 2012.

More recently, 2021, Eric Dietrich *et al* pinpoint outstanding problems:

> "AI researchers have not succeeded in solving the problems of machine mental semantics or the aboutness of computer symbols, they never seriously addressed the problem of machine consciousness, they never succeeded in getting machines to grasp what is relevant to what ... The frame problem remains unresolved, but lurks everywhere, unrecognised ... So, AI failed."[86]

The frame problem is a computational problem due to conditional combinatorial explosion. Semantic systems derived from the system's own experience don't run through a list of conditionals when deciding action, but simply, and quickly, follow existing connections.

Searle has contributed greatly to the project to develop genuinely intelligent machines by shining such a strong light on semantics, the essence of intelligence. But as argued, he shines it on the wrong machine, the Turing machine. The right machine, the electronic computer, has a different semantics.

I've argued that a place exists to start understanding semantics. This place is the outer-inner interface, the place where the inner reacts to the outer, the place where sensors react to the environment and create the elements of knowledge. The main process of the sensor is sensory transduction.

ChatGPT contains some semantic elements, but doesn't use them semantically. These elements are the connections (and nodes where they join). Acquiring knowledge is fundamentally the adding of connections, and ChatGPT doesn't add connections. And it doesn't have human-like sensory apparatus.

The next step is taking the elements of ChatGPT which are genuine advances, and abandoning the rest (such as back propagation and the predefined structurally unchanging ANN). If the temporal theory of intelligence outlined above is considered plausible then it could be tried. But whether plausible or not, AI needs to attack the problem of the intelligent computer from the semantic coalface, from the sensory interface, from the place where the elements of raw knowledge are manufactured.

If AI does this, perhaps the old mythologies can be cast aside and new research, semantic research, conducted under the original British title "Machine Intelligence". For as British AI pioneer Donald Michie pointed out:

> "The scientific goal of research work in artificial intelligence is the development of a systematic theory of intelligent processes, wherever they may be found; thus the terms 'artificial intelligence' is not an entirely happy one."[87]

After all, what genuinely intelligent machine would want to be called "artificially intelligent".

---

[86] Eric Dietrich, Chris Fields, John P. Sullins, Bram van Heuveln and Robin Zebrowski, (2021), *Great philosophical objections to artificial intelligence: The history and legacy of the AI wars,* London, UK: Bloomsbury, page 263.

[87] Donald Michie, (1974), *On Machine Intelligence*, NY: John Wiley & Sons, page 156.

8. CONCLUSIONS

My conclusions are that:

- ChatGPT doesn't understands online users' text questions because it doesn't get them.
- No text exists on the internet, so ChatGPT doesn't learn from internet text.
- The claim that ChatGPT learns from internet text reveals a fundamental flaw in AI theory.
- Many false claims are made about ChatGPT, some knowingly false. This mirrors the hyping of symbolic AI between 1950 and 1980 and will, as with symbolic AI, knee-cap progress towards AGI.
- AI doesn't understand semantics, but semantics (understanding, knowledge, meaning, intentionality) is the essence of intelligence.
- ChatGPT contains elements which, if used differently, would be semantic elements.
- Semantic content can be extracted from sensory data streams. A possible method and principles are suggested along with example code in C and assembler.

Released in late 2022, OpenAI's online computer chatbot ChatGPT amazed many with its apparent ability to understand text questions online users type. Experts and media alike said the chatbot was trained on text found on ("scraped from") the internet. Many said that from this training the chatbot learned the meanings of the text, and that this is why it understands, partly or fully, the text questions online users type.

Are these semantic claims true? A scientific answer seems crucial to assessing the extent of the advance ChatGPT represents. But no semantic analysis of ChatGPT has emerged. There seems only one semantic analysis of the computer, John Searle's famous 1980 Chinese room argument. Yet the semantic process of thinking, as everyone knows, is the essential process of intelligence.

Searle's Chinese room argument concludes that no computer could understand text. He assumes that computers are Turing machines. But are they? A semantic examination of the physics and chemistry of computer electronics shows they're not.

Computers internally manipulate electrons. No text is stored or manipulated in the electronic substrate which comprises computer electronics (and hence none is on the internet). There is no text in the communication channels between users' devices and the internet.

The only place text exists in computer systems is printed on the outsides of component such as capacitors and integrated circuit modules, and displayed on the exposed surfaces of keyboard keys, display screens and sheets of paper, put there so humans can see the shapes and interpret their meanings when using the machine as a tool.

To understand how the computer, in and of itself, could be intelligent, one needs to examine the science, the physics and chemistry, of the electronics. This shows that nothing in the operating electronics is text or images or sounds or anything apart from electrons.

Semantics should be at the forefront of AI research, but founder Turing excluded it from the scope of AI research, saying that the idea of a machine executing the process of thinking is "*too meaningless to deserve discussion*".

He couldn't explain the semantics of intelligence. AI has suffered ever since. AI's claims about ChatGPT being trained on text and understanding text are false. This misinformation is merely the latest example if AI's lack of understanding if semantics.

But the structure of ChatGPT's artificial neural network does contain elements which, if used differently, would play a semantic role. Such relational "virtual connections" can be extracted from sensory data streams and, after application of a repetition threshold, stored as forests of trees. This can create object structures and causal relations between them.

Virtual connections are valid relational connectives and can form the semantic basis of knowledge inside a computer, including knowledge of the meanings of the shapes of text.

9. REFERENCES

Searle, John R. (1980). Minds, brains, and programs. *The Behavioral and Brain Sciences*, (1980) 3, 417-457.

Searle, John R. (1984). *Minds, brains and Science: The 1984 Reith Lectures*. Cambridge, MA: Harvard University Press.

Searle, John R. (2014, 9 October). What your computer can’t know. *The New York Review of books.* https://nybooks.com/article/archives/2014/oct/09/what-your-computer-cant-know/

Searle, John R. (2018). "The Chinese Room Argument", Intelecom YouTube talk, 16 June 2018. https://m.youtube.com/watch?v=18SXA-G2peY.

Turing, Alan M. (1936). On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42), (published 1937), 230-265.

Turing, Alan M. (1946). Proposal for development in the Mathematics Division of an Automatic Computing Engine (ACE). In B. E. Carpenter and R. W. Doran (eds.), *A. M. Turing’s ACE Report of 1946 and other papers*, Cambridge, MA: MIT Press, 20-105.

Turing, Alan M. (1947). Lecture to the London Mathematical Society on 20 February 1947. In B. E. Carpenter and R. W. Doran (Eds.). *A. M. Turing’s ACE Report of 1946 and other papers*. Cambridge, MA: MIT Press, 106-124.

Turing, Alan M. (1950). Computing machinery and intelligence. *Mind*, 59(236), 433-460.

Turing, Alan M. (1951). *Programmers’ handbook for Manchester Electronic Computer Mark II*. Computing Machine Laboratory, University of Manchester. Photocopy courtesy B. Jack Copeland, annotated in Turing’s hand, errata sheet dated 13 March 1951.