Tattva – Journal of Philosophy
2024, Vol. 16, No. 2, 65-86
ISSN 0975-332X | https://doi.org/10.12726/tjp.32.5

An Ethics of Deconstruction in response to AI-generated Bias

Tanya Yadav* and Himanshu Jaysawal*

Abstract

The paper aims to respond to the ethical concern of biases generated by Artificial Intelligence systems. Even though biases enter an AI network via different channels, their presence in the algorithm can pose serious difficulties. AI systems have an algorithmic way of working where language is "formal," and meaning is "fixed." (Broussard, 2018, pp.31-39). We employ the deconstructive strategies of Jacques Derrida to understand the nature of this problem of AI bias through the examination of algorithmic/ programming language. Derridean philosophy looks at metaphysics as heavily dependent on notions such as logocentrism, where *logos* refers to the privileged part in a dichotomy. Logos is the *center of formal language,* which works as a system of signs. The main point of deconstruction is to apprise us of this privilege given to the "presence" of a concept or meaning over its "absence." Derrida's notion of *Undecidables* and *Aporia*, not only destabilizes rigid dichotomies like speech/writing but also give way to an opening of a concept in its 'impossible -possibility' (Anderson, 2012, p.75). In AI systems, programming/ algorithmic language conforming to its algorithm offers definitive answers to problems. This takes us a step ahead in the 'formalization' of language' (Beaney & Raysmith, 2024). Derrida, however, offers a response via his ethical deconstruction, where the process of 'completion' of any concept is deferred, and its meaning is 'undecided' (Roffe & Reynolds, 2004, pp. 37-47).

* Centre for Philosophy, School of Social Sciences, Jawaharlal Nehru University, New Delhi, India- 110067; tanyayadav105@gmail.com, himanshujaiswal369@gmail.com

Keywords: Algorithm, AI bias, deconstruction, logocentrism, *Undecidables*

Introduction

One would wonder why a French postmodern thinker and philosopher like Jacques Derrida is relevant in the most recent discussion on Artificial Intelligence and what it has to do with its ethical underpinnings. Derrida, a well-known twentieth-century figure who is famous for his unique strategy called 'deconstruction', has unpacked a new age of thinking, and analyzing language. His critique of Structuralism[1] in early works, *Of Grammatology* and *'Structure, Sign and Play in the Discourse of Human Sciences'* gave a fresh view on the categorizations and limits of language. The breakthrough in Derrida's works is not his critique of a *center-oriented* language which operates within a *sign system,* giving way to a *metaphysics of presence*; it is instead his capacity to acknowledge the errors in our reasoning. These errors in reasoning make us circle around stable, fixed centers of meaning, forming the illusion of the completion of a concept or an idea.

His concept of logocentrism[2] is an attack on the privileged role of logos, also regarded as the supposedly "ideal" part of the dichotomy that determines all the discourse, such as the position of speech over writing, thoroughly discussed in his Of Grammatology. It is also the case that this logos prioritizes a 'presence' over the 'absence' of the neglected term. In the binary opposition of speech/writing, presence marks speech over the absence (of the speaker) in writing. Subsequently an array of concepts gets associated with it, such as "phonocentrism," (Derrida, 1976, pp.11-12) the belief that a certain priority is given to "speech" over 'writing' in history. Similar to this is the biased conception of "ethnocentrism," where Derrida analyses that a certain superiority is given to "Western Man" over other human groups. (Glendinning, 2011, p.37). This is also evident in the binaries where male, light, truth, presence, and identity are regarded as superior and fundamental over female, dark, error, absence, and difference, respectively. Hence, logocentrism can be understood via various concepts through their search for "ideal meanings." Derrida also refers to this notion as "metaphysics of presence."

Now, it would be a mistake to look at logocentrism only as a way of critiquing language entrenched in the history of philosophy. What Derrida is instead attacking is a "logocentric way of thinking." Therefore, in the present context, the question that we must address is whether AI systems also function or are based on this sort of logocentric thinking. Is there a possibility of tracing these logocentric tendencies in the programming language of AI systems? Are there parallels between the idea of natural language[3] and programming language? What makes the language of AI systems an instrument of Derridean deconstruction? Finally, is there an ethics of deconstruction responding to AI-generated biases? Hence, this paper will be an attempt to examine an AI-generated problem through the lens of Derrida's deconstruction. The nature of study will be largely critical and thoroughly speculative.

AI-generated Bias

Why is AI important? Artificial Intelligence is used nowadays for smaller to bigger tasks, from choosing a restaurant that matches our food liking to checking on a patient's bone growth, accepting credit card applications, denying bail to a convict, and so on. In such tasks, AI makes decisions and provides recommendations. However, some of its recommendations may instead lead to biases or prejudices. For instance, predictive analysis, an AI tool that predicts criminals and the likelihood of crime in a particular area, discriminates against people of color when used by police in the US (Angwin et al., 2016). In another instance, Google ads recommended less-paid jobs to women when they searched online (Datta et al., 2015). AI biases can thus have a larger socio-political impact. The questions here are how biases enter the recommendations and decisions of AI systems and whether we can respond to this problem of AI bias. In order to explore these questions, let us first understand the functioning of AI systems and draw out the causes of AI bias.

Artificial Intelligence: Functioning

Artificial Intelligence is simply the possession of intelligence by a machine. It is a field that aims at building machines that can act and think like humans or act and think rationally (Bringsjord & Govindarajulu, 2018). AI systems are used for tasks like reasoning, problem-solving, natural language processing, social intelligence, and so on. To achieve such objectives, AI systems fundamentally implement programs. Such program implementation is achieved by syntactically manipulating symbols following instructions from the algorithm of that respective program. Programs are written in *assembly* language, also called programming language, and read in *machine language*. Meredith Broussard writes,

> Machine language translates symbols into binary so that the computer can perform calculations. Those symbols are the words and numbers that we humans use to communicate meaning to each other. It's a constructed system. The dialect you use to "speak" machine language is called *assembly language*. It assembles symbols into machine codes (Broussard, 2018, p.24).

Intelligence in AI systems is achieved primarily by manipulating symbol structures. Herbert Simon and Allen Newell, in their Physical Symbol System hypothesis, proposed that intelligent systems transform physical symbols to generate behaviour or intelligent actions. Thus, thinking or intelligence is the manipulation of symbols syntactically. Based on this fundamental idea, modern computers and AI systems are built (Bermudez, 2014, p.145).

Causes of AI bias

AI biases can enter a system during the designing, testing, or application stage. Based on this, it is supposed that there are three important sources of AI bias: "data, algorithm, and programmer" (Coeckelbergh, 2020, p.128). Let us look at each of them briefly.

Data

AI systems heavily rely on data for performance. They learn via feedback received from the users. However, the data they receive is not bias-free. Users are part of a society where biases are prevalent. When they enter the AI system, bias does not just get introduced in the system but also gets amplified. Apart from this, sometimes the data that is tested to run an AI app may not be representative and, when applied to a wider population, may also lead to bias. An example of AI bias entering via user data is the Google search results providing information on 'arrest record' when black (people) sounding names were searched. This did not happen when 'white-sounding' names were searched. This was because users in the past searched black names together with their arrest records. Machine learning recorded this and linked the arrest record of black people with the names of black people, so whenever such names were searched, it showed ads related to the arrest record (Silberg & Manyika, 2019). Another instance is when AI bias enters the system due to insufficient representation. Many image-recognition neural networks are learned via a popular dataset called the Image net dataset that has relatively acquired larger data from the US, leaving representation from other big countries like India and China, and hence leading to cultural bias (Zou & Schiebinger, 2018, pp.324-26).

Programmer

Programmers belong to the social milieu, and so their personal experiences, prejudices, and opinions can influence the programs they construct and thus lead to biased decisions in AI systems. One study showed that there was a lack of diversity in the groups of AI developers and data scientists. Most programmers were white men from Western countries within the age group of 20-40, and thus, it is probable that their viewpoint is dominant in decisions. This, in turn, affects the representation of marginalized groups such as people of color, old, disabled, women, and people from developing countries in forming AI-based decisions (Coeckelbergh, 2020, p. 128).

Algorithm

AI systems follow instructions in programming or algorithmic language. Sometimes, they may contain words that reflect historical or social inequalities, such as gender, race, or sexual orientation. Since AI systems merely follow the instructions and do not 'understand' the words they use, their implementation can lead to biased results. For example, Amazon hired a particular software whose algorithm used words like "executed" or "captured." The words are visibly male-centric, often found on men's resumes (Manyika et al., 2019). Thus, it hired more males than females in the company due to this fault in the algorithm. Though the system could have been trained for fair hiring, it gave biased results because the algorithm contained these words and lacked 'understanding' of these concepts the way humans do. Thus, the algorithm is a significant reason for the generation of AI biases. The algorithm also works easily on phenomena that are quantifiable. Systems get most data from their users on social media, who talk more about popular phenomena. Hence, there are chances that biased social phenomena gets transferred via users into algorithms,

> Algorithms are part of existing (biased) institutions and structures, but they may also amplify or introduce bias as they favor those phenomena and aspects of human behavior that are easily quantifiable over those which are hard or even impossible to measure. This problem is exacerbated by the fact that certain data may be easier to access and analyze than others, which has caused, for example, the role of Twitter for various societal phenomena to be overemphasized (Ntousti, 2020, p.3).

Do algorithms 'understand' bias?

In this section, we will explore how the algorithm is the major cause of the emanation of AI biases, and in the coming section, we will examine how Derridean deconstruction can be applied to the algorithmic nature and language of AI systems. We discussed that AI systems function on programs, and program implementation is simply symbol manipulation following the syntax of the programming language. Hence, intelligent actions are done through syntactic symbol manipulation. The reason why we view algorithms as the major cause of AI bias is because of their reliance on the 'syntax' of the respective programming language for any task (Samuel, 2022; Zou, J. & Schiebinger, L. 2018). Also, because it cannot grasp the meaning of the symbol structures or machine codes based on which it works. Let's see how this is so. Computer scientists generally propose that programming is sufficient for understanding or thinking. John Searle famously counterattacks this position. He argues that implementing a program or symbol manipulation is insufficient for understanding or thinking. His Chinese room thought experiment refuted the thesis of Strong AI, which says that computers can actually think or understand by merely implementing a program or manipulating symbols (Searle, 1980, p. 417). For Searle, there is more to thinking than just symbol manipulation,

> A digital computer is a device which manipulates symbols, without any reference to their meaning or interpretation. Human beings, on the other hand, when think, do something more than that. A human mind has meaningful thoughts, feelings, and mental contents generally. Formal symbols by themselves can never be enough for mental contents, because the symbols, by definition, have no meaning (or interpretation, or semantics) except insofar as someone outside the system gives to them. (Searle, 1989).

The Strong AI thesis makes multiple claims. Suppose a computer is given a story to read, and when asked certain questions, if the computer answers successfully, it can be assumed that it understood the story. If it was successful, then according to Strong AI, the machine would be able to think or understand. To understand, it just manipulated symbols following instructions in the program. Now, Searle proposes the Chinese room thought experiment to refute and reply to this Strong AI thesis. (Searle, 1980, p.418).

In the next experiment, a monolingual person sits in a room alone. S/he/they are given a rulebook containing instructions on manipulating Chinese symbols. Experimenters outside the room give her/him/them chunks of Chinese symbols. S/he/they are supposed to return those chunks of Chinese symbols by arranging them with respect to instructions given in the rule book. Experimenters outside the room first hand her/him/they, a story, then certain questions, and the subject/s return the answers. They are unaware that the first chunks of symbols given are stories and the second are questions; s/he/they arrange them by looking at their shape. People outside getting these answers guess that the person inside the room knows Chinese because her/his/their answers are at par with a native Chinese Speaker.

Have the person/s in the room understood Chinese? Clearly not. Searle proves that the person does not understand Chinese despite implementing the program, that is, following the rule book. In a similar manner, Searle argues, any computer or machine merely with the help of symbol manipulation cannot "understand" or "think" or grasp the meaning of symbol structures on which it works. (Searle, 1980, p. 418).

From the aforementioned example of the Chinese room argument, we can know that if biased words enter the algorithm of any program, then it will manifest in the decisions of AI systems. AI lacks the mechanism to neutralize the effect of the use of biased words because it can only operate and use those biased words without actually 'understanding' them. It cannot prove its decisions are discriminative towards certain people or groups merely by looking at them. AI does not have a mechanism to identify such prejudices or to correct them until and unless they are programmed in that manner.

In the above section, we learned the nature of AI systems and AI bias. We also criticized the idea, taking help from Searle's Chinese Room Argument, that AI systems do *think* and *understand* when they undergo a certain task. However, we contend that another argument to support this claim springs from Derrida's analysis of the "formalization" of language (Glendinning, 2011, pp.43-53). But to pose that argument, we must first establish how the formal nature of programming language connects to the formalization of natural or ordinary language, which Derrida vehemently critiques in his works. Thus, Derridean deconstruction that opposes the logocentric and totalizing nature of "formal" language will work with the same fervor against the totalizing framework of programming language.

The problem of language accuracy and fixity of meaning

Language is a symbol system. Through it, we communicate, share knowledge, think, command, etc. When philosophers of language worked towards forming a 'formal language,' they meant to eliminate any 'ambiguity' of meaning from words and instead assign somewhat of a mathematical certainty to them via a *formalized logical language* (Stroll & Donnellan, 2023). Analytic philosophers like Bertrand Russell attempted to analyze language and its underlying structure through reliance on rules of logic (Stroll & Donnellan, 2023, para. 29.). The motivation was early insights in Ludwig Wittgenstein's work *Tractatus Logico-Philosophicus* based on the idea that "the thesis that the structure of language mirrors the structure of reality has as a consequence that the meaning of a proposition is the particular fact to which it is isomorphic" (Stroll & Donnellan, 2023, para. 32). Russell's Logical Atomism runs along similar lines, where language is viewed as an aggregate of fixed and irreducible units like atoms, and there is a perfect correspondence between an "atomic proposition" in language to an "atomic fact" in the world ("Logical Atomism", 2012). Thus, what is evident from such popular inquiries into the nature of an "ideal formalized language" in philosophy, is the urge to grasp the structure of language in its totality.

System-builders in various disciplines have attempted to deploy "constructions", frameworks that help better understand the concepts and ideas involved. Words are deemed to have "fixed" and "accurate" meanings and language use is deployed in a similar sense. Wittgenstein's post-analytic work, *Philosophical Investigations*, enlightened philosophers about the difficulties of such a task. Here, Wittgenstein proposes that natural languages do not have any 'formal' structure, and their propositions or concepts do not have a fixed meaning; rather, the meaning of concepts is determined by how the concept is used in society. He revisited the functions of language and concluded that language is marked by a complex network of usages that give it sense and meaning. He also states that we are situated at intersections of various "language games" that determine language rules in that context (Wittgenstein, 1958, pp. 5, 82).

This realistic understanding of the limits of language and the impossibility of arriving at a "perfect" language for all times was furthered by Derrida's attack on the logocentrism of language in *Of Grammatology*. Derridean deconstruction has continually argued against the logocentric construction of meaning, which is at once portrayed as "ideal" and "pure". For instance, Derrida engages with the binary of speech/writing in his work where the *logos* is invariably associated with the "phonetic", spoken word over the written word,

> All the metaphysical determinations of truth, and even the one beyond metaphysical onto- theology that Heidegger reminds us of, are more or less immediately inseparable from the instance of the logos, or of a reason thought within the lineage of the logos, in whatever sense it is understood: in the pre-Socratic or the philosophical sense, in the sense of God's infinite understanding or in the anthropological sense, in the pre-Hegelian or the post- Hegelian sense. Within this *logos*, the original and essential link to the *phoné* has never been broken (Derrida, 1976, pp.10-11).

Thus, Derrida's deconstruction came off as a philosophical movement aspiring to critique all "totalizing" constructions in history, which are marked by *logos*. For Derrida, deconstruction differs from how the term came to be popularly known and used. It is not a method or a methodology to be applied to concepts that need revision. Instead, it is withdrawing from the very confinement of its meaning in any construct. *Logos* is the dominant privileged term in a dichotomy that goes on to be the *center* of the discourse (Derrida, 2004, pp.89-104). Deconstruction is an attempt to question this absolute authority of *logos* which renders other terms on the margins of inquiry. The logic of deconstruction is that it is a *double-dissymmetry* in its relation between a self and the other (Roffe, 2004, pp.37-39). J D Casten, in his work *Cybernetic Revelation: Deconstructing Artificial Intelligence*, suggests how deconstruction is, thus, "both 'about' construction and 'departing' construction,"

> The term 'deconstruction' could also be said to name the subject: naming subjectivity itself. But the subject and subjectivity are seen here, not in a full plentitude of self-presence—not a consciousness that is hooked up to a Logos of absolute knowledge handed down by Western philosophy. No, here subjectivity is temporal: its intentions, never fully worked out in advance other than in a possibly over-determined destiny projected from one's past into the future (Casten, 2012, p.466).

Applying it in the context of AI systems and specifically on the problem of AI bias, it will be intriguing to know how Derrida would respond to the logocentric nature of programming language. And for that, we would first have to investigate what exactly we mean by the logocentric nature of programming language.

Logocentric nature of programming language

Symbol manipulation systems

AI systems work with strictly rule-bound frameworks, using algorithms as the logic. The format for computational instructions is simply laid out via a specific algorithmic language or programming language best suited for that AI program. This stems from the task orientation of any AI system, ranging from the basic function of problem-solving, mathematical calculations, and game playing to more complex tasks of Natural language processing, which further enable a variety of real-world applications in the area of machine translation, speech recognition, chatbots, sentiment analysis, etc. As a computational procedure, NLP helps transform natural language data so that computers can 'understand' (not in the way a human understands) its meaning. (Gillis, 2024). Thus, it enables interaction between humans and computers in natural language. This interests us the most since the task of symbol manipulation and transfer of meaning happens at this stage.

Computation commands are based upon *"the conventions of the* syntax" of the chosen programming language (Meredith, 2018, p.16). Syntax is the rules for the functioning and relation of symbols. Hence, it forms the guide for symbol manipulation. Such syntactic manipulation is the manipulation of symbols and is carried out via *algorithms*. Same in the case of machine language,

> One very common computer alphabet is the binary alphabet {0, 1}. The symbols in the binary alphabet can be combined into strings of 0s and 1s that are the "words" of the computer language. These are the programs hard-wired into the computer and written in what is usually called machine language (Bermúdez, 2014, p.143).

Thereby, it is difficult for AI systems to grasp a semantical understanding of concepts for their sole dependence on syntactical arrangement. This results in the assignment of 'fixed meanings' to words and symbols used in algorithmic language, which becomes problematic whenever we encounter a multiplicity of meanings in differing contexts. For instance, high-level computer language and the same natural language would grasp the meaning of the following two statements differently. The statement "the child is in pen," when contrasted with another statement, "the ink is in pen," would process two different meanings of the word 'pen,' first as an enclosure and second as a writing implement from the lens of natural language (Hauser, 2023, para. 28-29). For a computer, however, both statements would refer to a single meaning of pen as a writing instrument because this meaning is fixed, corresponding to its instructional machine code. Now, if this machine code is sophisticated with several meaning options, the machine would still choose the most probabilistic one based on semantical connections (Landaeur et al., 1998, p. 260). This proves how machines are far from 'knowing' the meaning of symbols they use, and algorithms are far from 'understanding' the context in which they are used.

Algorithmic language differs from natural language for it aims to convey unequivocal messages that adhere to a 'stable' meaning, devoid of ambiguities. However, the formal nature of programming language or algorithmic language makes direct links to the formal 'ideal' language which has been a dream of analytic philosophers (Preston, n.d.). The former focuses on the rules of algorithm, while the latter focuses on the rules of logic. Both are formalized in that they refuse to factor in 'context' and other contingencies, so social phenomena like biases and prejudices get dangerous when codified in their systems.

In the past, Derridean deconstruction has argued against the possibility of an "ideal language", (Preston, n.d.) by continually stressing the problems with the logocentric nature of language that leads to phallogocentric and ethnocentric biases. We notice that the functions and nature of programming language make it logocentric in its commitments, too. In that sense, a certain *presence* is privileged here too. In Derrida's opinion, metaphysics has always prioritized purity of *presence* over the contingent and complicated. (Reynolds, n.d.). The algorithmic nature of programming language makes it inevitable for its concepts and meanings to rely on 'presence,' 'clarity,' 'consistency,' 'stability', and 'fixity.' The symbols adhere to formulaic *understandings*, deemed as 'complete' and viewed in 'totality.' In other words, programs do not have loose ends in meaning as that can obscure instructions.

The idea of *logos* is such that it falls on the dominant and privileged side of the dichotomy, which then decides the discourse for the marginal and neglected concepts. Apart from speech/writing, Derrida focuses on other binaries of mind/body, signified/signifier, presence/absence, and so on (Singh, 2023). In the case of programming language, we function with the seeming binary oppositions of syntax/semantics, formal/informal, rigid/flexible, accurate/ambiguous, certain/uncertain, and so on. And it is evident how far one side of the binary is preferred and prioritized in the 'construction' of AI systems.

Algorithm got no 'free play'

In another essay, Derrida thoroughly analyzes concepts like *signifier* and *signified* while critiquing Structuralism (Derrida, 2004, pp. 89-104). The latter is a school of thought that understands language as a system of signs, wherein a signifier refers to the word or its sound image, and a *signified* is the concept or meaning linked to that word. (Derrida, 1982, p. 10). Such linguistic signs are regarded as arbitrary as they make sense to us, only within the "structure" of that language. In this essay, Derrida exposes the contradictions that plague structuralism, which searches for a fundamental structure that can act as a "center" or a "fixed" origin; this origin is supposed to provide anchoring and stability to the production of meaning, knowing well that language does not have any fixed relation with reality (Smith-Laing, 2018, p.11).

For Derrida, this knowledge rules out the idea of any *fixed* center. Thus, meaning is always subjected to the "free play" of concepts. (Derrida, 2004, pp. 89-104). By acknowledging the *absence* of a center with any natural function in language, he arrives at a perspective that multiple equally valid vantage points are called discourses, and thus, there is no "transcendental signified" to be grasped, giving way to *infinite play* of meaning (Derrida, 2004, pp. 90-91). For instance, a signifier (word) such as ‘rose’ does not represent a signified (concept) transparently and fully. It could mean a flower, evoke the emotions of love and romance, symbolize movements, or associate with thorns and pain, depending upon its context and cultural usage. Thus, a signifier does not refer to any external meaning (residing in a “transcendental signified”) but refers to other signifiers (Singh, 2023).

Hence, the concept of “play” looks for the meaning of a signifier, which indefinitely gets *deferred* and delayed in an endless chain of linguistic references. (Singh, 2023). Derrida, by his concept of “infinite play”, suggests the difficulty of arriving at a ‘stable’ meaning or center (Singh, 2023). Now, focusing on the algorithmic concepts that constitute a programming language, we can easily recognize the *logocentric patterns* of assignment of meaning. Let us try to understand this through the presence/absence dichotomy, where algorithms again rely on the *metaphysics of presence*. AI concepts are definitive by ‘what they are’ and not by ‘what they are not’. This means that we grasp them by the ‘presence’ of a definition and not by the ‘absence’ of its numerous possible meanings in other contexts. Unlike the “*trace*” in Derrida, which is both a rupture and a movement for a signifier towards another signifier, where meaning is not confined by mere ‘presence’, but is rather deferred by ‘absence’ (Derrida, 1982, p. 23). In a language system of *signs*, definitions are nothing but a result of *simplification*, a process of arriving at a stable and fixed meaning of a *sign*, ignoring its *free play* (Singh, 2023).

Similar is the nature of algorithms, as they are based on ‘stable’ meanings of symbols which the AI system adopts owing to its predictability. Predictability of the meaning of a particular signifier gives us confidence in its usage and application at all times (Bermudez, 2014, pp. 154-155). For instance, we need to arrive at a ‘stable’ definition of the word ‘dog’ to reach other complex problem-solving levels, like examining whether dogs are faithful companions of humans. Thus, the requirement for a definitive and fixed meaning is the most immediate in the case of artificial systems. This highly limits the programming language to engage in any sort of *free play* in conceptual understanding.

Non-contextual logic of the algorithm

When a programmer assigns the meaning to a word or a sentence that the machine reads in terms of symbol structures, the meaning assigned is non-contextual in the program. However, when the same word is used in natural (ordinary) language, the meaning of the word/concept is variable and contextual. Even though the programmer can intentionally or unintentionally let contextual contingencies factor in the formation of programs, such as in the case of programmer's bias, we still consider computer programs as immune to context owing to the scientific nature of technology, which argues for objective neutrality.

Non-contextuality characterizes the 'rigid' nature of programming language based on syntactical rules of formal logic, which are inflexible and constant in character. The immobile nature of programming language can also render some computational procedures redundant over time, while natural languages continually revisit their tenets. For instance, words like savage, serving, mothering, etc. have acquired supplementary 'sense' in which they are used today; apart from their literal meaning, they now have pop-culture associations where they are popularized as slang. However, programming language and its instructions steer clear of the complexity of context. Derridean deconstruction attacked logocentrism because it enables the illusion of "completeness" of any concept (Singh, 2002, pp.51-54). Derrida believes that such totalizations in language are useless and impossible as there is an infinite play and slippage of meaning. The idea of a stable center is, thus, a myth. Context plays a significant role for Derrida, as evident in his famous statement, "There is nothing outside the text" (Derrida, 1988, pp.136-37). He meant that the formation and application of meaning are exercised within a context that marks the language, its grammar, cultural preferences, and a web of other meanings.

Algorithms can only be contextual in the limited sense where they are best suited to perform problem-solving, calculating, natural language processing, etc. However, their language and its use are codified and programmed when defining the fundamental concepts. Since programs cannot undergo frequent revisions in their conceptual meanings and understandings, they go on with a single preferred meaning for symbols. Hence, they, too, further the illusion of 'complete' ideas and concepts, then *formalize* them in an algorithm.

Can deconstruction respond to AI-generated bias?

Finally, we will investigate the problem of AI bias and whether deconstruction is a useful response to it. Our aim here is not to provide solutions to the problem of bias in AI systems. However, the idea is to take a closer look at the problem itself and, with the help of Derridean deconstruction, show how logocentrism in AI problematizes and further assists the continuation of such biases.

AI biases are discriminatory stances resulting from decision-making or knowledge generation at the end of AI systems; it mainly has three causes: data, programmer and algorithm. Derrida's deconstructive strategies have already exposed biases and contradictions prevalent in Western philosophy, such as phallogocentric bias (where the male perspective is privileged over the female perspective), ethnocentrism (where Western man is regarded superior to other groups of men), and so on. Deconstruction as a movement has always overturned the binary oppositions in favor of the marginalized concept, subverting dominant ways of thinking. Hence, an ethics of deconstruction aims at a certain fairness by rendering its subject free of biases and hierarchical thinking. The question then arises- can these ethics of deconstruction respond meaningfully to AI-generated biases? With the help of deconstruction, we can investigate the logocentric rationale for a) entering of biases and b) retention of biases in AI systems.

Entering of bias in AI systems

Per our earlier stand, Artificial Intelligence holds 'intelligence' not in the sense humans do; at best, it can manipulate symbols syntactically. Biases can enter the system through different channels. However, the framework of programming systems and the nature of algorithmic language, which can easily include partial/biased understandings, are often ignored. Because AI systems merely process information and give results governed by syntax rules, the focus is not on the semantics. It barely understands the contingencies of social and cultural meaning assigned to words or concepts; bias originates from such partial understandings. For instance, suppose the concept of a doll is to be programmed in an AI system, and when asked to define the concept of the doll, the system wrongly identifies it as a girl toy. This biased understanding of the term ‘doll’ can be a result of gender roles prevalent in society. However, the AI system will not recognize its biased nature, neither when it comes from the programmer, nor when it is received from user input. And not even via algorithms since they cannot ‘understand’ biases or be sensitive to that level of information. Value judgments are not encoded or embedded in algorithms (Samuel, 2022). Dangerously enough, though, it can solidify those biases and foster them by giving recommendations and decisions.

Retention of bias in AI systems

After a bias has entered the system, it becomes even more difficult to identify and rectify it. The logocentric nature of programming language makes it hard to dissipate the binary oppositional understanding of concepts such as syntax/semantics, formal/informal, and accurate/ambiguous. This results from the fixity and inflexibility of meaning. Hence, biases unknowingly become embedded in these symbolizations until the programmer identifies and corrects them. One major obstacle in removing biases is the assumption that AI systems cannot have any. The popular opinion that ‘machines don’t go wrong’, which is a biased human understanding, makes us ignore such biases without realizing it. Programs are not created in a social vacuum. When dealing with large amounts of data, dominant trends can influence AI systems, resulting in social biases creeping in.

An ethics of deconstruction is supposedly not working outside the context; it is, in fact, taking inspiration from within. When we acknowledge the logocentricity of AI systems, we can move towards ensuring that fairness is prioritized, whether it is in equal representation of concepts or obstructing the inflow of all biases. The problem of logocentric language is that it communicates meaning as a ‘fixed’ entity. There is seldom a possibility of ‘play’ and, thereby, revision of any computational concept. Even if one gets such chances to revise AI frameworks, one cannot ignore the place of a programmer. Often programmers have a limited understanding of relevant values owing to their own privileged position (Samuel, 2022).

Derrida's Concepts of the Undecidable and Aporia

As a critical movement, deconstruction has been credited with exposing the inconsistencies and inherent biases within formalizations of structures and systems. However, other important conceptions in Derrida's work can be used as a response to logocentrism. These are termed as *undecidable* and a*poria*. The former is a conception that, as a third category, seems neither present nor absent, and, in this way, it challenges the dualism of dichotomous oppositions of presence/ absence, inside/outside, and so on (Reynolds, 2004, p.46). The name *undecidable* marks its simultaneous possibility and impossibility, but a conscious move to not arrive at a decision. The same can be said of *aporia*, which are paradoxical moments whose condition of possibility is also the condition of their impossibility, such as in the case of gift, hospitality, forgiveness, etc. (Reynolds, 2023).

We cannot recommend the applicability of these open-ended concepts for AI systems, nor will they help mitigate AI biases. However, there is something intrinsically ethical in these conceptions which is also reflected in the motivation of deconstruction as a technique. It is their conviction to stay away from oppositional thinking or any formalization that claims to arrive at a 'purity' of meaning. The idea of a 'center' in logocentric thinking is influenced by Western thought's obsession with the idea of truth, one origin, absolute principle, God, and so on. (Singh, 2023, para. 23). With the advent of artificial intelligence, one would not want the return of this 'center' under the name of mathematical certainty, fixed meaning, context-neutral, algorithmic, etc.

Conclusion

The paper attempts to deploy Derrida's deconstruction strategy to respond to the problem of AI bias meaningfully. For this, the paper provided parallels within the 'formalization' of natural language and programming language that fosters biases due to 'fixity' of meaning and lack of 'understanding'. Derridean deconstruction is used to investigate how programming language is not untouched by logocentrism. Free play of concepts becomes impossible due to rigid meaning, formal nature, and predictability. AI systems are unaffected by a dichotomous understanding of concepts. Hence, deconstruction's imperative role in blurring such boundaries suggests an overhaul of the logocentric way of thinking.

Endnotes

[1] "Structuralism worked on the basis that words are only related to reality by linguistic conventions. There is, for instance, no actual link between the word "book" and a real book except for the fact that English-speakers use that word to refer to that kind of object. To think of it another way, the word "book" can only mean what it means to us because of its place in the "structure" of English as a whole language, and our familiarity with that structure. Most importantly, we know that "book" has a different meaning from other words, and this, in fact, is how we know "book" refers to the kind of object it does. We know it does not mean the same as "paper," "pamphlet," "block," "scroll," and so on, because it is, in various ways, different from all of them. Our knowledge of English allows us to understand how it differs from those words, and so to understand each other when we say or write it. Because of this, structuralism argues, all meaning comes from webs of difference." (Smith-Laing, 2017, p.10).

[2] Derrida describes how this heritage of logocentrism in Heidegger and Hegel reiterates the privileging of 'presence.' "This notion remains therefore within the heritage of that logocentrism which is also a phonocentrism: absolute proximity of voice and being, of voice and the meaning of being, of voice and the ideality of meaning. Hegel demonstrates very clearly the strange privilege of sound in idealization, the production of the concept and the self-presence of the subject…We already have a foreboding that phonocentrism merges with the historical determination of the meaning of being in general as *presence*, with all the subdeterminations which depend on this general form and which organize within it their system and their historical sequence (presence of the thing to the sight as *eidos*, presence as substance/essence/existence [*ousia*], temporal presence as point (*stigmè*] of the now or of the moment [*nun*], the self-presence of the cogito, consciousness, subjectivity, the co-presence of the other and of the self, intersubjectivity as the intentional phenomenon of the ego, and so forth). Logocentrism would thus support the determination of the being of the entity as presence. To the extent that such a logocentrism is not totally absent from Heidegger's thought, perhaps it still holds that thought within the epoch of onto-theology, within the philosophy of presence, that is to say within philosophy *itself*." (Derrida, 1976, pp.11-12).

[3] What we mean by natural language here is the human languages (emphasis on English language), and the term is used specifically to contrast it with the 'artificiality' of machine-based languages.

References

Anderson, N. (2012). *Derrida: Ethics under Erasure*. Continuum Publishing Group.

Angwin J., Larson J., Mattu, & Kirchner L. (2016, May 23). Machine bias. *ProPublica*. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Beaney, M. & Raysmith, T. (2024). Analysis. In E.N. Zalta & U. Nodelman (Eds.), *The Stanford Encyclopedia of Philosophy* (Fall 2024 ed.). Stanford University. https://plato.standford.edu/entries/ana lysis/s6.html#3

Bermudez, P. (2014). *Cognitive Science: An Introduction to the Science of Mind*. Cambridge University Press.

Butler, C. (2002). *Post-Modernism: A Very Short Introduction*. Oxford University Press.

Bringsjord, S. & Govindarajulu, N S. (2018). Artificial Intelligence. In E.N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Fall 2024). Stanford University. https://plato.stanford.edu/entries/artificial-intelligence/

Britannica, The Editors of Encyclopaedia. (2012, May 4). *Logical Atomism. Encyclopedia Britannica.* https://www.britannica.com/topic/Logical-Atomism

Britannica, The Editors of Encyclopaedia. (2017, June 20). ideal language. *Encyclopedia Britannica*. https://www.britannica.com/topic/ideal-language

Broussard, M. (2018). *Artificial Unintelligence: How Computers Misunderstand the World*. The MIT Press.

Casten, J D. (2012). *Cybernetic Revelation: Deconstructing Artificial Intelligence*. Post Egoism Media.

Coeckelbergh, M. (2020). *AI Ethics*. The MIT Press.

Datta A., Tschantz M C. & Datta A. (2015). Automated Experiments on Ad Privacy Settings: A Tale on Opacity, Choice, and Discrimination. *Proceedings on Privacy Enhancing Technologies*, 92-112. https://a`rxiv.org/abs/1408.6491

Derrida, J. (1976). *Of Grammatology*. The John Hopkins.

Derrida, J. (1982). *Margins of Philosophy*. The Harvester Press.

Derrida, J. (1988). *Limited Inc*. Northwestern University Press.

Derrida, J. (1993). *Aporia*. Stanford University Press.

Derrida, J. (2000). Structure, Sign and Play in the Discourse of the Human Sciences. In D. Lodge & N. Wood (Eds.), *Modern Criticism and Theory: A Reader.* (pp. 89-103). Pearson Education Limited. (Original work published 1966).

Derrida, J. (2005). *The Politics of Friendship*. Verso.

Derrida, J. (2006). *Spectres of Marx*. Routledge.

Foster, S.R. (2005). Causation in Antidiscrimination Law: Beyond Intent versus Impact. *Houston Law Review*, *41*(5), 1470-1548. https://ir.lawnet.fordham.edu/faculty_scholarship/188

Glendinning, S. (2011). *Derrida: A Very Short Introduction*. Oxford University Press.

Gillis, A S. (n.d.). What is natural language processing (NLP)? *TechTarget*. Retrieved August 2024, from, https://www.techtarget.com/searchenterpriseai/definition/natural-language-processing-NLP#:~:text=How%20does%20natural%20language%20processing,way%20a%20computer%20can%20understand

Hauser, L. (n.d.). Artificial Intelligence. *The Internet Encyclopedia of Philosophy*. https://iep.utm.edu/artificial-intelligence/#SSH4b.ii

Holland, N J. (n.d.). Deconstruction. *The Internet Encyclopedia of Philosophy*. https://iep.utm.edu/deconstruction/

Kreines, J. (2015). *Reason in the World: Hegel's Metaphysics and Philosophical Appeal*. Oxford University Press.

Landauer, T. K., Foltz, P. W. & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes, 25*(2-3). 259-284. https://doi.org/10.1080/01638539809545028

Manyika, J. & Silberg, J. (2019, June 6). Tackling bias in artificial intelligence (and in humans). *McKinsey Global Institute*. https://www.mckinsey.com/featured-insights/artificial-intelli gence/tackling-bias-in-artificial-intelligence-and-in-humans

Manyika J., Silberg J. & Presten, B. (2019, October 25). What do we do about the biases in AI? *Harvard Business Review*. https://hbr.org/2019/10/what-do-we-do-about-the-biases-in-ai

Ntoutsi, E., Fafalios, P., Gadiraju, U., Iosifidis, V., Nejdl, W., Vidal, M., Ruggieri, S., Turini, F., Papadopoulos, S., Krasanakis, E., Kompatsiaris, I., Kinder-Kurlanda, K., Wagner, C., Karimi, F., Fernandez, M., Alani, H., Berendt, B., Kruegel, T., … & Staab, S. (2020). Bias in data-driven artificial intelligence systems—an introductory survey. *WIREs Data Mining and Knowledge Discovery*, *10*(3). https://doi.org/10.1002/widm.1356

Preston, A. (n.d.). Analytic Philosophy. *The Internet encyclopedia of philosophy*. https://iep.utm.edu/analytic-philosophy/#H3

Reynolds, J. (n.d.). Jacques Derrida. *The Internet Encyclopedia of Philosophy*. https://iep.utm.edu/jacques-derrida/

Reynolds, J. & Roffe, J. (Eds.). (2004). *Understanding Derrida*. Continuum.

Samuel, S. (2022, April 19). Why it's so damn hard to make AI fair and unbiased. *Vox.* https://www.vox.com/future-perfect/229-16602/ai-bias-fairness-tradeoffs-artificial-intelligence

Searle, J. (1980). Minds, brains and programs. *Behavioral and Brain Sciences, 3*(3), 417-457. https://doi.org/10.1017/S0140525X0000 5756

Searle, J. (1989, February 16). Artificial Intelligence and the Chinese Room: An Exchange. *The New York Review*. https://www.nybo-oks.com/articles/1989/02/16/artificial-intelligence-and-the-ch -inese-room-an-ex/#:~:text=His%20argument%2C%20called% 20sometimes%20the,were%20thrown%20back%20and%20forth

Singh, P. (2023, May 24). Derrida's Structure, Sign and Play-Summary and Analysis. *Literature and Criticism*. https://www. literat-ureandcriticism.com/structure-sign-and-play/#:~:text =Derrida's%20Structure%2C%20Sign%20and%20Play%20refe rs%20to%20the%20structural%20center,fixed%20origin%20 and%20an%20end.

Singh, R.P. (2002). *Philosophy: Modern and Postmodern*. Om Publications.

Smith-Laing, T. (2017). *An Analysis of Jacques Derrida's Structure, Sign and Play in the Discourse of Human Sciences*. Macat Library.

Stroll, A. & Donnellan, K.S. Analytic Philosophy. In *Encyclopedia Britannica*, Retrieved September 25, 2023, https://www.britannica.com/topic/analytic-philosophy

Wittgenstein, L. (1922). *Tractatus Logico-Philosophicus*. The Edinburgh Press.

Wittgenstein, L. (1958). *Philosophical Investigations*. Basil Blackwell.

Zou, J & Schiebinger, L. (2018, July 12). Design AI So That it's fair. *Nature.* https://www.nature.com/articles/d41586-018-05707-8.pdf