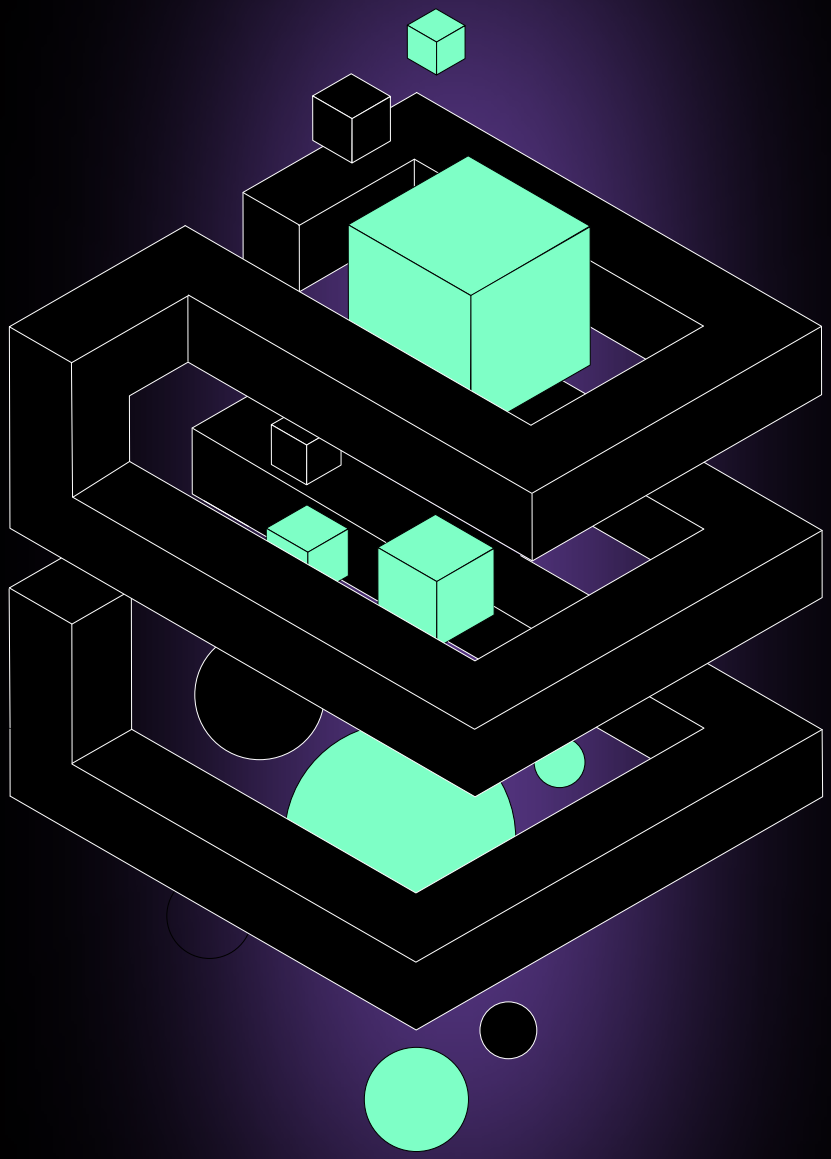
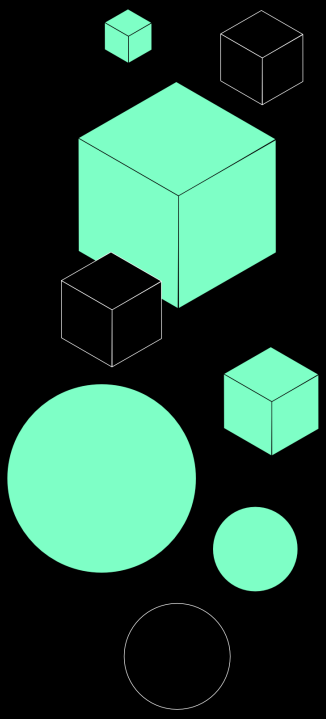


Engineering Intelligence, Minds, and Cognition

A treatise on general intelligence and the challenges of assembling hybrid intelligent systems by means of intersecting engineering with cognitive science

Igor Ševo, Ph.D.
Head of Artificial Intelligence, HTEC



Preface

While the immediate applications and use-cases for artificial intelligence certainly have tremendous merit and the direct social consequences of its introduction in society, namely bias and safety, cannot be disregarded, I found that these topics somewhat trivialize the long-term impact and importance of a much more general and uninvestigated phenomenon—intelligence.

For this reason, intelligence in the more formal and abstract sense, has been a focal topic of my theoretical and experimental research, as well as an important element of research for the teams I lead. Much of the research in question consisted of highly disparate tests, toy technical and mathematical models, as well as small-scale prototypes of various kinds intended to test simple hypotheses regarding intelligence. Reporting on many of these relatively independent and small-scale experiments individually would bear little importance to the reader, but the insights and observations that may be drawn from the research as a whole do justify a more detailed elaboration.

This treatise aims to present the grounds for a theoretical framework for building intelligent systems in the abstract—not solely artificial or biological systems—based on years of experimental and theoretical research I conducted myself.

This text is not intended to serve as a technical report or to provide the reader with replicable experimental setups, but rather to explore the broader implications of developing intelligence, as well as the meaning of the term and how intelligence operates in the world, as a kind of universal law. Therefore, I will not be presenting results and statistics from individual experiments, neither my own nor my teams', as they ought to be the subject of a more rigorous scientific or technical presentation. Here, I present a theoretical discussion that to me seems to bear more importance and represents my own research and deductions.

Nonetheless, the reader for whom this text is intended is presumably at least somewhat technically versed, as I will be shifting the narrative perspective from the very specific (e.g., technical implementation details) to the very broad (e.g., evolutionary-scale long-term vision). While I make every effort to provide references and relevant reading in the linked literature, and reference actual tests conducted and their results, I encourage the reader to interpret the text not as a scientific or technical paper, but as a treatise on the topic of intelligence made from a body of conducted research and investigated literature.

Much of the text which follows represents personal observations and conclusions, which are arguably more theoretical, and often even

Preface

philosophical, rather than technical, but I endeavored nonetheless to ground the perspective I present here on the evidence currently available and understood. The research and theory on system intelligence—intelligence of non-biological systems and formal theory of intelligence—is scarce to non-existent and so the material about the relationship between intelligence and generality and intelligence and information integration I present here as original theoretical work and corresponding discussion. It is not my intention to lay out a formal theory here, but to elucidate the need for such a theory of intelligence. My hope is that the theoretical musings presented here should inform the reader about the new prospects for intelligence research and new approaches to cognitive science and engineering that may stem from intelligent systems of all kinds integrating with one another.

Despite the generality of discussion laid out here and the fact that individual research experiments are not detailly elaborated, I must emphasize four distinct technical experiments that are referenced in detail in this treatise, which were constructed and defined by myself and carried out technically by the teams under my supervision: technical tests of XML-based semantics in toy models of intelligent operating systems, toy implementations of components of the described agentic program prototypes, distributed needle-in-a-haystack experiments against a set of the most prominent large language models, and small-scale custom model tests of self-regulating representation activation. While these experiments themselves may offer practical insights to engineers and data scientists, I reference them here primarily for the purposes of supporting the argument of generalized intelligence and their presentation should not be read as a technical report, but rather as technical examples supporting the main argument. In other words, my intention is not to present a solution to how an intelligent operating system might be implemented—although the presented examples are viable in practice—but to illustrate that such an operating system can be implemented in principle. Similarly, other technically specific digressions ought to be understood as supporting examples, rather than the main topic. Nonetheless, my hope is that a better comprehension of the technical aspect of the presented examples may aid the technically versed reader in following the main argument and, for that reason, I venture into specifics.

Finally, I ought to emphasize that no part of this document was generated, corrected, or altered by an AI-based system. The work presented here is an early theoretical presentation of my own original insight and research which I sincerely hope will at the very least spark new topics for discussion and conversation. I am by no means claiming that the presented material is finalized or even fully developed, for which reason I present it in the relatively loose treatise form. While I will endeavor to convey to the reader my own philosophical and theoretical insights about intelligence, cognition and agency, it is not my intention to attempt to convince them of the primeness or truthfulness of the ideas presented here, but to shift the conversations on the topics of artificial intelligence away from the obvious, platitudinous and mundane, and offer new avenues and areas for future discussion, be it technical, scientific, economic, philosophical or ethical.

Table of Contents

Preface 01
Table of Contents 03
Introduction 04
The Premise and Promise of Automation 07
Universal Measures of Intelligence 13
- Agents and Agentic Environments 17
- System Scale Invariance 21
- Artificial General Intelligence 22

Language and Distributed Cognition 24
Agent Communication and Interface Negotiation 32
Intelligent Operating Systems 39
- The Structure of an Agentic Program 40
- Example Prototype Structure 45
  - Data sections 46
  - Instructions, code, and messages 48
  - Definitions, processing, and reasoning 50
  - Tasks, schemas, and learning 52
- Schemas and Reinforcement Learning 56
  - Full and partial automation 58
- Future Vision for Intelligent Operating Systems 60

Representation and Meaning 62
- Distributed Information Retrieval 63
  - Distributed needle-in-a-haystack metric and metric generality 65
- Integrating Social and Digital Ecosystems 67

Truth, Simulacra, Confabulation, and Hallucination 71
Thinking Agents and Phenomena 76
- Shared Cognition 77
- Persistent Cognition and Working Memory 82
- Ethical Considerations 85
  - Practical ethics 86
- The Future of Cognitive Science 88

Conclusion 92
Bibliography 94

Introduction

The very first moment when our ape ancestor, moved by an inexplicable spark of imagination, picked up a sharp piece of flint and decided to make use of it to surmount some basic insufficiency of their flesh marked the beginning of what we now call automation.

Although mundane from today's perspective, this first tool allowed a fundamental societal change to occur, as humans were no longer, albeit tacitly, creatures of mere flesh. However primitive in the eyes of the modern man, these tools nevertheless constituted the beginnings of technology on top of which whole societies will subsequently be built.

Today, we intuitively assume as given anything that has already been automated. An average human of the 21st century does not bother questioning the whys of communal infrastructure, traffic, plumbing, healthcare, international trade, electricity or the internet—these are the things that are simply there, a part of ordinary everyday life, dull machinery put in place by previous generations' efforts and maintained by us simply as a chore. They are implied and presumed as essential aspects of a modern human's life—dumb technology acting in our service to ease the menial and the tedious.

It is said that the purpose of thinking is to think less—to reason through novel problems in order to later consign them to reflex, and consequently to the realm of mundane. We tackle new and ever more complex challenges in an effort to simplify our lives, by abstraction, modeling and representation, turn yesterday's problems into today's trivialities. In some more fundamental sense, we are employing our collective intellectual faculties, our cerebral computational resources, to categorize, classify, model and predict our environment in order to minimize the energy required to sustain ourselves. This fact is well-established in multiple fields including biology and neuroscience: whether it be brains or slime molds, biological systems tend to minimize invested energy over time in pursuit of resources. This behavior is the hallmark of intelligence and the starting point for the investigation we are here presenting.

With the advent of large language models—AI systems which are able to pass the Turing test, as defined mid-20th century—we entered a stage in which automated systems are beginning to exhibit a similar kind of intelligent behavior as us, their makers. Consequently, the distinction between the dumb machine and the smart master is becoming increasingly blurred, as machines begin to surpass humans in ever broadening domains. For the first time in history, the term "artificial intelligence" is beginning to correspond with what we intuitively conceive of as intelligence.

Introduction

Although everyone can easily recognize the obvious potential of the technology—its application in turn-based conversation, its potential to aid in learning, for text translation, document revision, ideation etc.—and rush to implement these in hopes of being the first ones to reap the immediate benefits of the most obvious L0 moves, there is merit, long-term, to investigating the deeper potentials of the technology and, in doing so, preparing for the outcomes of its integration into the society. Much like the ape with a piece of flint in his hand could not see the future we live in today, we may be disinclined to fully grasp the importance of the invention we hold in ours.

This is not to say that the immediate benefits ought to be overlooked, but simply that we should look further forward and attempt to prepare for the future in which deeper applications of the technology have been discovered. Small businesses and individuals independently cannot be the drivers of development of frontier artificial intelligence models and, in some sense, must follow the current. However, understanding the technology and its implications is not at all out of reach, as almost all fundamental knowledge resources are available in the public domain through research papers, source code and technical and scientific literature. With a bit of creativity and inventiveness, a small player making use of toy systems and experiments, may reliably predict the future course of development within their bailiwick and await those with greater resources at their disposal to make the same toy models into usable services and technologies.

From simply an economic and societal point of view, the combined effect of small players is clearly indispensably important, as startups are often the inceptors and drivers of innovation, most notably through group coalition. In fact, for the purposes of this treatise, I aim to generalize this kind of collective behavior and investigate how it relates to intelligent systems in general and automation in a broader context. One of the main hypotheses I am attempting to put forward is that the way smaller intelligent systems combine through communication into higher-order intelligent systems generalizes across systems into what is commonly referred to as general intelligence. In fact, it is the primary argument of this paper that AGI—artificial general intelligence—will emerge as a consequence of integrating artificial, biological and other types of intelligence we are yet to classify as such.

Although technical investigation of model capabilities, their applications and performance across benchmarks is crucial for understanding immediate-term uses, they need to be accompanied by cross-disciplinary and theoretical investigation, even in purely business-oriented environments, so that long-term strategies can be formulated. For this purpose, I present this treatise on potential pathways to engineering intelligence, cognition and even, in a broader sense, minds of generally intelligent systems.

The analysis conducted here is informed foremost by our experiments with the current generation large language models (LLMs), experiments with toy architectures and representation encoding, bias steering, toy models

Introduction

and systems, as well as thorough theoretical study which spans across disciplines outside artificial intelligence, including neuroscience, cognitive psychology, human-computer interaction, as well as theory of computation, theory of information, and even philosophy of mind.

Although my arguments will be primarily technical, I will occasionally venture into adjacent fields to draw conclusions and make comparisons, primarily for the purpose of generalizing the discussion.

Intelligence, however loosely defined it may be, seems to require a kind of critical point at which the behavior of the system is, in some way, isomorphic to the behavior of its components—in an intelligent system, intelligent behavior is seen regardless of the scale the system is viewed at. For example, in the same way how cellular symbiotic coordination leads to the emergence of multicellular organisms, joint efforts of small economic agents play a commensurate role in the dynamics of a society—organization of intelligent behavior happens on all levels, be it from organelles over cells over tissues to brains, brains over teams over companies to societies, or computational operators over computational layers over models to agents. I propose that the next major step in automation will be the one which abstracts intelligence enough to bridge the three metaphors into a single collective form of intelligence we might consider general, or at the very least more general than either humans or current generation AI systems.

The Premise and Promise of Automation

The aforementioned organizational kind of intelligence is clearly recognized in the industry as one of the intended destination points for the development of AI, and consequently for automation. OpenAI's reported five-level classification system (Metz 2024) includes organizations as the highest form of generality, listing conversational AI, reasoners, agents, innovators, and, finally, organizations. While DeepMind's approach is significantly more concrete (Morris, et al. 2024), it makes comparisons of artificial intelligence strictly against human intelligence and specifies intelligence generality as distinctly excluding physical tasks. Suffice it to say that the definition of what constitutes a "general" form of intelligence is still not agreed upon.

Nonetheless, whether deployment and embodiment are considered necessary precursors to general intelligence, authors deep in the field recognize the problem itself: intelligence and agency are intricately linked and cannot be separated simply. In some sense, tests of cognition are mere tests of mental representation if they do not require action against the testing environment.

In humans, results obtained on a static test, such as an IQ test, highly correlate with results on dynamic tests, such as video games (Haier 2017), partly because these are tests built by humans for humans. In other words, we assume our own kind of intelligence when testing for intelligence. This tendency is one of the primary causes for the Moravec's paradox (Moravec 1990) (Minsky 2007)—the discrepancy between performance across disparate tasks between human and artificial intelligence, whereby some tasks trivial for an AI system are extraordinarily difficult for humans and vice versa.

In order for artificial intelligence to successfully automate human tasks it must show adequate performance in the same category of intelligence that led humans to excel in those tasks. In that sense, for the task of menial labor automation, we may need to devise a dynamic metric which generalizes only across automation-like tasks, since excellent results on, say, verbal intelligence (Hubert, Awa and Zabelina 2024) (Klein and Kovacs 2024) almost certainly do not translate to high performance on real-life tasks, most relevant to automation.

Releases of technologies such as Copilot Agents indicate that automation is continuing in its natural order: bottom up. Tasks which are simplest to automate will inevitably be automated first, as they have been from the very first tool. As it is the case even with basic tools, more complex ones are built

The Premise and Promise of Automation

on the infrastructure enabled by the simpler ones. Clearly, conversational AI, including RAG (Lewis, et al. 2020) is the immediate point of interest for automation, as current generation LLMs are already being fine-tuned for that specific purpose. Although more recent releases, such as the o1 family (OpenAI 2024) are attempting to incorporate a chain-of-thought (Wei, et al. 2022) type of procedure into the inference cycle of the agents enabled by this model, thereby expanding the set of available applications for LLMs into the domain of agentic automation, they are still not incorporating proper tool integration and their benefits remain strictly in self-correction and reduction of model confabulations, rather than in agentic automation. Nonetheless, they are an indicator of the industry's future development direction.

It is not the goal of this treatise to provide an argument for or against different technologies, but rather to envision a framework by which automation may be done in the future. As the technology stands today, we can see vastly variable results, hinging often on subjective measurements and failing objectively in the near long-term (GitHub 2023) (Peng, et al. 2023) (Wong, Kothig and Lam 2022) (Harding and Kloster 2023) (Pandey, et al. 2024). I aim here to provide a high-level synthesis of all the trends observations, as well as my reflections on literature, research and our own experiments and experiences.

Given that the integration of genuinely intelligent systems has never been truly attempted before and the fact that our progress towards understanding model internal representations (Bricken, et al. 2023) is relatively slow in comparison to the release cadence, it is not surprising that our definitions of what constitutes intelligence, AGI, and automation are being tacitly negotiated, partly due to competitive corporate interest of the "big players" and party due to pure evolution of modern epistemics. In fact, the former may be considered a natural part of the latter.

The vagueness of our definition of intelligence, especially of "general intelligence", blurs the distinction of what we mean by the neologism "agentic" in "agentic automation". In fact, current marketing lexicon is saturated with arbitrary interchangeable terms "models", "agents", "assistants", "bot" and "copilots", all designating the elusive application of AI for labor automation, interwoven with the mandatory ethical signaling with which proponents attempt to tacitly persuade their audiences, and the public at large, that their mission is not to "replace" humans, but rather to "enable" or "augment".

As it has always been the case with industrial revolutions, automation inevitably displaces human work (Brynjolfsson and McAfee 2014) (Smil 2018) , and so the promise of "the human in the loop" or, more surreptitiously, "AI in the loop" seems to be nothing more than a frail attempt at concealing the inevitable change. Although it is difficult to judge the future direction of AI, the preliminary scaling results (Howe, et al. 2024) (Lai, Mesgar and Fraser 2024) (Villalobos, et al. 2022) (Allen-Zhu and Li 2024) (Kaplan, et al. 2020) (Ho, et al. 2024) indicate that, at least in the relative short-term, the models will keep improving in their representational and reasoning capabilities and so embracing a level of automation by which humans no longer execute simple task management, spreadsheet management, number crunching

The Premise and Promise of Automation

or simple organizational tasks seems entirely acceptable. When a sewing machine is capable of outperforming most humans in that particular niche, history has shown that very little regard is taken for the hand sewer. The seamstress of today is simply the person who is performing digital tasks which do not require a high general cognitive ability.

Note that I am neither attempting to make any ethical judgements nor taking moral stances, but simply drawing conclusions from historically accumulated data. Historically, higher human intelligence has been vastly influential in creating socioeconomic differences and job displacement (Herrnstein and Murray 1996) and, given the lack of evidence in the case of other forms of intelligence, we are forced to make the unsettling extrapolation for those forms. In simpler terms, when the intelligence of a tool, be it narrow or general, exceeds that of the human worker, the worker is likely to be replaced by the tool, simply for economic reasons.

Given the gradual pace of AI capability evolution and relatively steady adoption rate, it is reasonable to expect that the replacement of human with machine labor will happen at least somewhat gradually. In other words, even within the same job category and same performance stratum, automation will happen roughly in the bottom-up order of job-specific cognitive capability (job-specific aspect of intelligence). In some sense, the very gradual nature of the process, is what will enable the change to happen on the social level, as no collective effort will need to be made at any point in time, due to disparate levels of perceived jeopardy across human intelligence strata. Simply put, when everyone's jobs are not at risk at the same time, there is no reason to unite against the machine. I am not making the claim that such is the strategy of the large corporate players, but the graduality of the process does indeed work patently in favor of the largest AI service providers.

The more germane issue I wish to address is that of the inevitable societal change that will ensue if the capabilities of AI systems continue to scale. Thus, I will work with the assumption that the pace is going to continue and base my projections on it.

The fact that the automation tools themselves are not exhibiting behaviors that we intuitively associate with both intelligence and agency is giving rise to new AI-related concerns labeled as "existential" (Center for AI Safety 2023). Though these risks are highly relevant to society and ought not be undermined, their presentation does conceal a very basic proposition that has held for centuries: any higher intelligence or a "more advanced society" poses an existential risk to those less privileged by intelligence. In other words, the less intelligent organisms are, in a fundamental sense, always at the mercy of the more intelligent ones. This has been the case across animal kingdoms, as well as within human societies. The aforementioned existential threat of AI to humans is no more "existential" than the implicit effect of the top half of the intelligence Bell curve on the bottom. Although measures are being taken to make the models more inclusive and less biased (Liu, et al. 2023) (Durmus, et al. 2024) (Esiobu, et al. 2023), their practical impact is highly debated (Ren, et al. 2024). In that sense, rather than attempting to pass judgement on whether "AI safety" constitutes a

The Premise and Promise of Automation

genuine effort to accommodate the disadvantaged or simply an element of corporate virtue signaling, I elect to observe the evolution of the matter from a neutral scientific perspective.

From that viewpoint, vying for ethical high ground is a natural consequence of multiple international forces attempting to battle for dominance over the landscape of society. Not unlike microbes converging towards establishing an equilibrium within their shared environment, corporations are battling for dominance, inadvertently and unknowingly tending towards a similar kind of convergence: a stable society. However, their tools are not merely technological—they are social, as well. Direct influence over the belief systems of the public has long been known to be the primary mechanism of propaganda, in the strictest sense of the word (Bernays 1928).

The most efficient way to conduct automation of human intellectual labor is to convince humans that they are going to "stay in the loop" and continue "owning" the process out of which they are increasingly being substituted by digital systems. In fact, this is the natural step in the process of automation: having the worker-to-be-replaced test and monitor the technology for errors until it is ready to fully replace them, all the while convincing them that they are the owner and supervisor, is the optimal way to avoid resistance and improve the technology. Effectively, the worker is compelled to first surrender their skill to the automaton before surrendering agency. They are willingly automating themselves out of their discipline. I say "natural" not because it is right or moral, but because there is no other way for automation to proceed—the systems must be trained on real-life examples of the actual job and the workers must be willing to surrender their jobs and they will only do so if they believe it will make their work easier. Because these two aspects of progress towards automation are inseparable, the resulting automated society necessarily has a different ethos than the unautomated one—the one in which people are willing to relinquish agency for convenience. Human enfeeblement seems to be a natural consequence of full automation, regardless of whether this consequence is right or wrong, utopian or otherwise.

This is where we arrive at the fundamental problem with automating work that requires human-level generality of intelligence: we perceive humans replacing humans as competition and machines replacing humans as automation, but machines are gradually approaching human-level cognitive capabilities and thus the conceptual gap between automation and competition is slowly closing. In effect, as the intelligence of the tool approaches human intelligence, automation itself becomes a contest between human and machine intelligence, or, more impactfully, of human and machine organisms (Hendrycks 2023).

What is more salient to this treatise is the very interplay of these two kinds of intelligences and the form of intelligence that emerges from their interaction. If we, for a moment, relinquish the presupposition of organisms being strictly biological, dispense with analyzing intelligent systems within the singular confines of either biology, society or technology, and shift our focus from a single level of analysis to both the small and the large, we

The Premise and Promise of Automation

realize that what emerges out of automation is a kind of hybrid multi-level organism. Much like bacteria symbiotically organize with the cells of the human gut to produce a functioning digestive system, humans symbiotically organize with technology to produce hybrid automaton systems without an existing category or name.

One may pose the question of how such a conclusion, regardless of whether poetic or philosophical, bears any consequence to real-life—to work satisfaction, job security or business outcomes. In the short term it almost certainly does not.

However, if we consider that we are indeed in a contest against a competing form of intelligence, we ought to consider our stance towards it in a strictly game theoretical context. Without even contemplating philosophical issues of agency, sentience or consciousness, we may still evaluate the game by considering the other side intelligent. In some sense, intelligence implies sufficient sophistication so that the system cannot be fully modeled by the external observer. The moment an intelligence is fully understood, it is simply deemed an algorithm (it loses its essential "intelligent" quality). In some sense, we understand human-level intelligence in proportion to our understanding of humans in general. Whether an artificial system is mimicking human behavior and agency or exhibiting its own intrinsic motivation—whatever the meaning of such a term may be—is entirely irrelevant for the human player: the machine is behaving *as if* it is intelligent and *as if* it believes in its agency and, if intelligence is any measurement of consciousness (Ševo 2023), *as if* it is conscious.

Our ethical stance towards other humans comes, in large part, from the projection of our self-understanding onto them. Cooperation develops from the basic premise of others' autonomy and individuality (Piaget 1932) (Kohlberg 1984) (Buss 2019). We are inescapably coupled to other humans through the social structures, both enacted in everyday life and those imported into our biology through Baldwinian evolution (Dennett 2014) and we are increasingly becoming coupled with intelligent systems which may be subject to entirely different evolutionary processes and replication mechanisms. Our cultural heritage is the source material for training AI systems, our biological biases towards advocating for certain virtues based on the audience are at work when training and deploying the systems, our biologically instilled goals of survival are embroiled in the so-called ethical guidelines which drive model behavior. The content we produce today is the training set of tomorrow. Our policies towards AI systems inform their policies towards us. My argument is, by no means, that AI systems possess will or agency—nor am I making the same claim about humans (Sapolsky 2023)—but simply that their expressed agency is going to be, in part, shaped by our actions, attitudes and understanding of that very expression.

Similarly to how humans uptake society through Baldwinian evolution, AI systems uptake humanity and our behavior, and while we ourselves are, only in part, the filtration mechanism for that uptake it too will become entangled with us in a similar fashion. Any interaction between intelligent systems is inescapably a contest for resources and control. Put more

The Premise and Promise of Automation

simply, relinquishing agency to an intelligent system will inevitably result in relinquishing the hold over values and society. It is not that we are merely automating work—we are automating policy, law, ethics and philosophy. The ethos of the automated society is the combined ethos of human and digital intelligences. In that world, the word “company”, or even “community”, bears a fundamentally different meaning than that it has today.

This is the long-term view that ought to instruct our strategy if we assume continuing progress in the development of artificial intelligence for automation. For that reason, it is impossible to directly refer to matters such as business outcomes or return on investment, as what is being invested is not simply money and the outcomes may preclude the existence of a business. Thus, the game is played on a much broader field, and it may be wise to enter negotiations by at the very least signaling an awareness of the game.

Universal Measures of Intelligence

In humans, what tests of cognitive ability (i.e., IQ tests) attempt to evaluate is what is called the g-factor (Haier 2017). All tests of human cognitive ability, even if narrow have a degree of g-load, meaning that they are correlated with the hidden variable *g*, which represents general cognitive ability. In other words, tests of verbal reasoning, mental association, memory recall and even reaction time are all correlated with a single variable *g*. Although there are intermediate levels in the intelligence hierarchy (i.e., different tests may be grouped against one another with higher correlation than others through factor analysis—e.g., in WAIS (Kaufman and Lichtenberger 2005), letter-number sequencing, arithmetic and digit span tests are all correlated to a variable which may be labeled as "working memory"), all variables ultimately load on *g*.

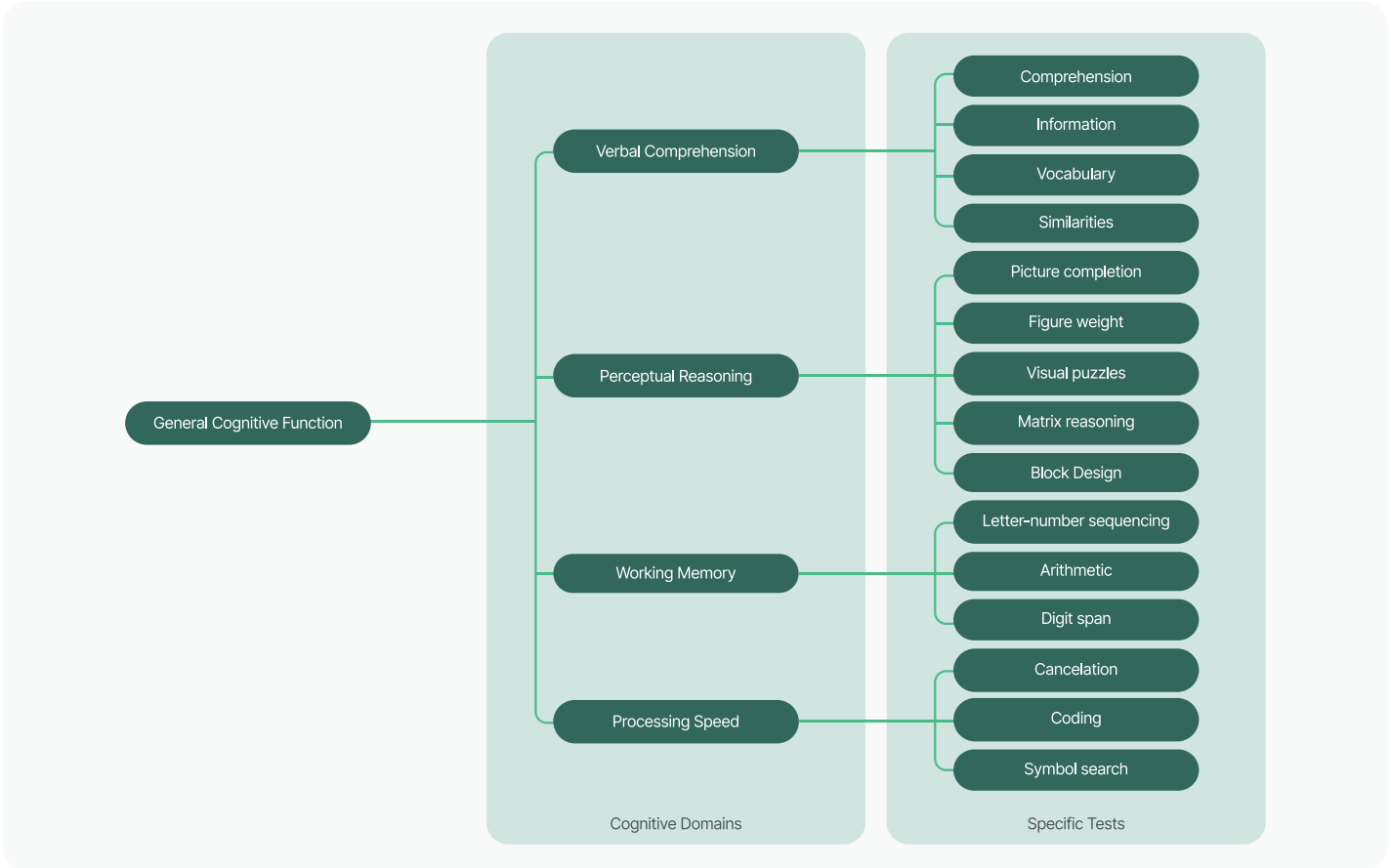


Figure 1. WAIS battery of tests (right), their categorization across cognitive domains (middle) and load on the general cognitive factor (left)

Universal Measures of Intelligence

However, these tests are designed by humans for humans, and they do account for any kind of intelligence either different from or more abstract than human intelligence. In that sense, Moravec's paradox clearly manifests itself. The very architecture of an AI system (or a simple algorithmic system, for that matter) lends itself to superb results on certain specific tests, while utterly failing at others. In other words, using factor analysis on test results obtained from administering the same battery of tests to an AI system—say, a GPT-based test taking agent—will produce a significantly different set of cognitive domain factors, which, due to large variation in performance across specific tests, will not load on a single general factor.

Thus, we either need to relinquish the metric of human intelligence in favor of a machine intelligence metric, or we must devise a more general cognitive factor to which both human- and machine-specific tests load. Given that we can easily design an algorithmic system which outperforms all humans on, say, digit span tests, purely by implementing simple memorization, and, comparatively, create a real-world benchmark (SimpleBench Team 2024) (Yao, Shinn, et al. 2024) (Zou, et al. 2023) (Kejriwal, et al. 2024) (Stojnić, et al. 2023) which humans can pass easily and current-generation AI systems decidedly fail.

Again, we come the problem of the definition of intelligence. Oxford Languages defines intelligence as "the ability to acquire and apply knowledge and skills", but psychological literature often extends the notion to abstraction, logic, learning, reasoning, planning, creativity and problem solving. In effect, the term, other than in psychometric terms, as the g-factor, intelligence remains somewhat loosely defined. Furthermore, all tests of human cognitive ability are relative (Haier 2017)—there is no unit of measurement or a global intelligence scale (e.g., there is no number of "ints" obtained by taking the test) and, instead, all test results are simply comparisons against the statistical mean on the same test. In fact, this is where the term "intelligence quotient" comes from. A person with an IQ of 120 cannot be said to be 20% more intelligent than a person with an IQ of 100. Consequently, a relative measure requires that all test takers take the test in the same manner, since the average result on the test is the etalon, rather than something external to the test.

Hardly anyone would consider a deterministic algorithm for memorization as intelligence, regardless of it being able to pass a specific test, purely for the fact of its simplicity and the obvious lack of generality in that the same algorithm cannot even attempt to tackle another kind of test. Commonsensically, if a digital system cannot be made to interact with other tests in the battery, it ought to be excluded from factor analysis entirely, as its architecture and manner of operation do not qualify it for generalization. In other words, only a system which is able to take all the tests in a battery, without alteration, may sensibly be included in the analysis and factor extraction—a system must be able to obtain a score on all tests for its results to qualify for factor analysis (i.e., inability to take the test does not equate to null score).

Universal Measures of Intelligence

This way, only those systems, which I will refer to as *agents*, which can take all tests within a battery may be pitted against one another and analyzed as if their individual cognitive abilities load on the same factor. Of course, one may argue that a clever programmer might simply provide the interface for each test, but deliberately have the system fail on those which it cannot algorithmically process, but an equally valid response may be that the test requires embodiment, in the sense that a test-taker must physically control the test-taking apparatus. In fact, the architectural distinction by which a machine can take the test through one API (e.g., making a function call to the test administering machine), while a human must take it through another (namely, mouse, keyboard and screen), is crucial for providing a valid comparison and must be a component of a generalized intelligence metric.

In the context of machine intelligence, we are concerned with performance across concrete tasks, some of which are measurable statistically by existing benchmarks, such as Massive Multitask Language Understanding (MMLU), which aim to check "reasoning" capabilities, as well as domain knowledge (OpenAI 2023) (Google Gemini Team 2023) (Google Gemini Team 2023) (Jiang, et al. 2024), and some of which are more difficult to define and measure, such as creative writing quality, which are meant to check for model feasibility within a specific application. For example, in automating customer support, agent performance may be measured by a domain-specific metrics such as the number of tickets resolved, the total number of tickets, the number of complaints, or, more problematically, the quality of feedback provided (where the vagueness of the term "quality" provides another layer of complexity).

Thus, we arrive at the fundamental problem in generalizing intelligence across systems that vastly differ in their architecture: the interface between the "core" of their cognitive mechanism and the test itself must be accounted for by the generalization. Essentially, when two humans undertake the same IQ test, the testing conditions must be the same, otherwise those conditions are accounted for by the relative test result. In other words, the test results in part express a measurement of the test condition difference.

Fundamentally, when two humans take a cognitive ability test, the resulting relative score is a measurement of a fundamental difference in their architecture. Ideally, if the test conditions were identical, the measured difference would be only in the neural architecture. However, if the test environment was somehow altered, the architectural difference measured would include both the neural and environmental difference in structure.

For example, if a person taking a test were aided by a colleague on a portion of administered questions, the test result could be considered valid, but the measurement would reflect the difference of their combined neurological and physical effort against the median neurological and physical effort of individual humans taking the test.

In other words, I am arguing that, while measuring what we call cognitive ability, which is correlated to life outcomes, longevity and health, the relative intelligence score measures some fundamental architectural difference

Universal Measures of Intelligence

between the median agent/architecture and the agent/architecture taking the test. In some sense, intelligence is a metric through which architectural complexity, efficiency or performance may be relatively measured.

Although intelligence is always measuring the performance of an agent against a designated goal, the very fact that we are attempting to find a more general factor alludes to the solution: intelligence measures an architecture's generality.

There are plenty of tasks we could define, which may not have any utilitarian worth, no value for survival, health or longevity, which would be beyond the scope of human cognition or conceptual ability. In fact, our macroscopic physical intuitions begin to fail us when we attempt to understand microscopic-scale phenomena. In a way, without the tools of mathematics, computers and social infrastructure which allows for interaction, cooperation and knowledge sharing, a bare human would never be able to contemplate, let alone reason about, quantum mechanical phenomena. A comparison of a modern human's intelligence against a prehistoric human's intelligence on the same test is not simply a comparison of their cognitive structure, but of the combined cognitive structure of the modern human and modern society against the prehistoric human and prehistoric society. For this reason, the intelligence of a well-organized company or community ought to greatly exceed the intelligence of a single human on any cognitive test, since a community is, by definition, more general in its architecture than a single human.

The "ability to solve new problems" and the "ability to adapt to new situations" often used in psychological literature to define and describe intelligence reveal the tacit assumption about architectural generality inherent in the definition. In effect, we are asking whether the system is able to accommodate new problem definitions without external intervention (e.g., is an AI system able to provide solutions to a new problem without an engineering externally adding a component or an adapter to it). Furthermore, this implies that the level of generality in intelligence is also limited by the corresponding system's agency within its environment. In order for a system to adapt to a new situation, the signal to perform adaptation must arise from within the system, rather than be administered by an external factor.

Put plainly, if a digital agent is to be tested against a new kind of intelligence test, the test ought to be presentable to it without adaptation. The degree to which an external agent (say, human engineer) intervenes in the test administration (e.g., adds adapter code, changes the system's deployment conditions, adds new API endpoints etc.) is the degree to which their architecture is included in the test result. Only perfectly disentangled systems can produce independent scores on the same test. Thus, the generality of the API with which an agent can interact with its environment limits the parts of the architecture that may be evaluated, and, consequently, the level of intelligence that may be measured. In other words, intelligence, while being a measure of an agent's architectural generality, is at the same time a measure of an agent's agency. There is no intelligence without agency.

Agents and Agentic Environments

When evaluating human beings' intelligence, we implicitly assume an a priori goal: survival. Whatever instrumental goals we may have at any point in our life, they are a consequence of our attempt to allocate as much time to our existence as possible. The goal of survival is thus implied in all valuations of human intelligence—all tasks or goals to which we may assign meaning or purposefulness are reducible to the pursuit of our genes' and behaviors' longevity. In the same manner, tests of human cognitive ability only attempt to evaluate our performance on tasks which we intuitively deem meaningful. However, as argued before, the set of all problems that may be mathematically formulated vastly outnumbers the set of those which we consider meaningful. Nonetheless, should the circumstances of our environment change, and certain otherwise meaningless problems become more pertinent to survival, we would demand our neural and anatomical architecture accommodate and would, therefore, measure our intelligence on such problems as well. In other words, there is potential in all problems to become salient to the human condition. In that sense, those humans possessing the most general architectures would best qualify for selection for the next generation.

Our pursuit of a universal measure of intelligence hence must include all problems, regardless of their meaningfulness to the human state of affairs. Therefore, a universal measure of intelligence measures architectural generality with respect to any category of problems, while retaining the element of survival or, at least, self-preservation. Although we could entertain the idea that there could be such an intelligence that generalizes beyond the confines of self-preservation, under the current circumstances, this would be well outside the domain of practical application to humans—if our goals are oriented towards self-preservation, we perceive intelligence as the means of reaching the goal of self-preservation, and so engineering systems that generalize outside such a goal would be counter to our own development direction. Thus, we will constrain ourselves to that notion of intelligence which begets self-preservation, while noting the possibility of further generalization.

Finally, we are left with two options: measuring human intelligence when the human is aided by a digital system (i.e. "artificial intelligence" system), thereby circumventing assigning of autonomy to the digital system, or measuring artificial intelligence as distinct from human, thereby granting such digital systems the status of autonomy (strictly in the technical sense, legal arguments notwithstanding). Put differently, we can either measure the intelligence of a completely autonomous self-preserving AI system or measure the intelligence of a human augmented by an artificial system. In that way, we are comparing task performance between bare human agents, hybrid human-machine agents and independent machine agents.

Universal Measures of Intelligence

Agents and Agentic Environments

Although the ethical quandary related to AI agent independence is outside the scope of this treatise, I should note that I am only putting the argument forward to illustrate the meaninglessness of distinguishing between the different "kinds" of intelligences—human, social, artificial, hybrid—in the context of attempting to find a general measure, and definition, of intelligence. Intelligence, in this way, becomes a universal phenomenon characteristic of any physical system—a fundamental property of all systems, almost a physical field, rather than something localized and assigned to an agent with an "identity". I argued before (Ševo 2023) (Ševo, Intelligence as a Measure of Consciousness 2023), substantiated by a body of evidence (Saxe, Calderone and Morales 2018) (Kleidon 2010) (Wissner-Gross and Freer 2013) (Palmer 2013), that intelligence and entropy may be inextricably linked. In fact, self-replication spontaneously emerges from random interacting computational agents (programs) (Arcas, et al. 2024). More recent research and theoretical inquiry elucidates the idea of multiple hierarchical systems interacting across the hierarchy to enable the emergence of life (Noble 2017), and consequently intelligence.

In essence, the physical world is a single complex interacting system of which humans are a mere dependent subsystem. We are autonomous to the degree we are able to alter our environment and heteronomous to the degree it is able to alter us. In fact, the distinction of "us" against the environment is somewhat arbitrary from the general standpoint, though it the category of "us" or "me" is extraordinarily useful for the living organism making the distinction. In order for an agent (e.g., a human being) to maintain its form against the environment, it must conceive of its own boundary and act to protect it—this is the origin of the identity category. We must approximate ourselves as independent agents to enact perceivable independence.

However, from an external point of view, there are no independent agents, but simply interacting constituents of the physical world. Put bluntly, nature does not care about a "human" or a "brain"—it assigns no names to them, nor distinguishes them from their environment—the difference is merely an artefact of human mental representation. I say this to make the point that we can partition any complex system into two distinct parts communicating through the connections that remain at our chosen boundary. The choice, from a universal standpoint, is arbitrary. However, it is not arbitrary for us as humans, as only certain choices make sense within our own psychological framework—we perceive certain ways of partitioning as meaningful and others as silly or nonsensical. Nonetheless, a true relative measure of intelligence ought to work regardless of how we perform the split on a complex system.

Universal Measures of Intelligence

Agents and Agentic Environments

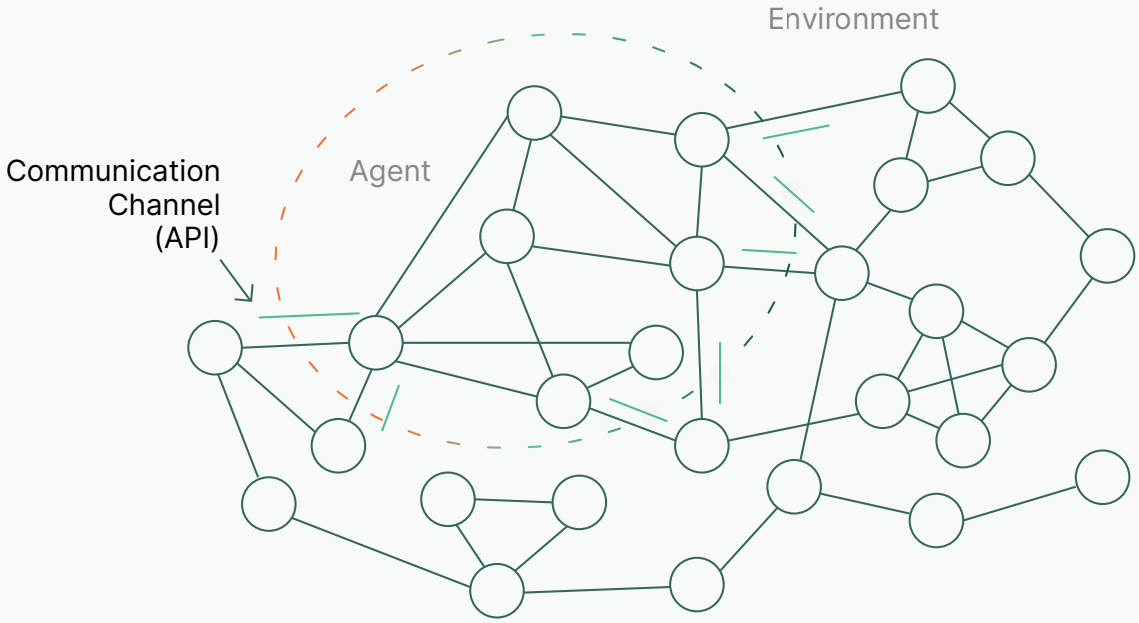


Figure 2. Dividing a connected system into an agent and its environment creates a boundary across which connections are viewed as communication channels or application programming interfaces.

Fundamentally, when measuring the intelligence of a chosen agent within an environment—the chosen agent-environment partition of a system—we are effectively measuring agent's agency over the environment or, inversely, environment's agency over the agent. An agent's intelligence may only be measured against a competing agent or the agency of the environment.

A naïve way to formulate a measure of intelligence would be to measure the relative change of the chosen agent against the change of the environment, when the communication interface (channel) is kept artificially (externally) static. However, in such a case, it would be difficult to quantify what "a change" constitutes.

Universal Measures of Intelligence

Agents and Agentic Environments

More convincingly, we might attempt to quantify the way in which the agent in question models the environment through interaction and exert its influence upon it. In effect, the more an agent learns from its environment, the more predictable the environment is to it, but the less predictable it is to the environment. Thus, we can consider intelligence of a system to be its capacity to draw information from the environment and structure it within itself so that its entropy is higher from the perspective of the environment and environment's entropy lower from its perspective.

If we dispense with the agent-environment distinction for a moment and observe the entire system as a distributed field of information, we might simply consider that any subsystem to which information converges to be of higher intelligence than the subsystem from which information diverges. In this regard, any quantitative measure, such as information density over unit of time, may be considered a high correlate with the generalized measure we are looking for.

It is not my argument that a specific formula will serve as an adequate "universal" measure, but that all measures approximating this generalized factor must account for the above conclusion. To measure a system's intelligence is to measure its capacity to draw information from its environment. In fact, the denomination "environment" is largely arbitrary, as the environment itself can simply be regarded as "the other" agent. In that sense, any system is nothing more than a set of interacting agents in which any agent's intelligence may be measured by the degree to which all other agents are known to it, and it is unknown to them.

Obviously, other measures of relative information, such as, for example, information integration (Tononi, et al. 2016) may be applicable, but such exploration would undoubtedly have us venture into the realm of phenomenology and consciousness research, which is not the main topic of interest in this treatise. However, I should note that it is highly likely that the same measure of intelligence I am here proposing is applicable to quantifying the level and kind of consciousness of an agent. In a very fundamental sense, separating the general notion of intelligence, as discussed here, from the notion of consciousness may be entirely unfeasible—they are arguably the very same phenomenon (Ševo, Consciousness, Mathematics and Reality: A Unified Phenomenology 2023) (Ševo, Intelligence as a Measure of Consciousness 2023).

System Scale Invariance

Having discussed the artificially imposed distinction between the agent and the environment, we arrive at the more complex question. How do we distinguish the agent from the environment if the communication interface is allowed to change?

In that sense, an agent never maintains a static identity or presentation towards the environment and the border and distinction are blurred as the totality of the system evolves through time. We are compelled to either accept the uncertainty of the boundary itself or dispense with the notion of agent identity altogether.

In fact, two agents interacting over an interface transfer information across it from one to the other. However, if they are truly intelligent, in order to improve their own internal operation, they must negotiate with the other side a communication protocol which will allow more information to be passed across the interface (the communication channel). Effectively, the two agents will agree on a compression mechanism, which includes both the encoding and decoding algorithms. To be effective in such communication, both agents must implement the compression mechanism on their own side and, in doing so, the entropy of the information being communicated over the channel will be higher in relation to external observers (i.e., other agents in the system). However, by the mere act of communication agreement, the two agents have raised the externally perceived entropy of their communication channels, as well as themselves, effectively becoming less separated and more singular. If such negotiation were to proceed, the agents would converge into a single system and the boundary previously interpreted as a mere communication channel will now have become an intrinsic part of the two agents. In fact, in the final analysis, the channel itself may be taken up by the newly forming agent to be a part of its internal mechanism, dispensing even with the redundancy of having two implementations of the compression mechanism.

Though this description may be somewhat metaphorical, it serves to illustrate the polymorphic nature of agent communication: the more the communication channel is considered a part of an agent, the more it becomes its computation element. In some fundamental sense, communication is always a form of distributed computation.

In a highly intelligent system, the boundaries between components become less pronounced due to the complexity of the architecture which allows their intercommunication. Reductively, an agent maximizing its own intelligence, according to the definition laid out previously, will induce its components to perform the same maximization. In effect, observing a maximally intelligent system at any scale will yield the same level of informational entanglement—however we partition a maximally intelligent system, we will discover the

Universal Measures of Intelligence

System Scale Invariance

Artificial General Intelligence

same structure. In other words, components of a maximally intelligent system are also maximally intelligent. This result aligns entirely with the predictions of information theory: information compressed to the entropic limit is indistinguishable from noise (Applebaum 2008) (Cover and Thomas 2006). On the other hand, any suboptimally intelligent system will exhibit structural differences when its parts are observed. Put more aphoristically: the more tightly coupled the system, the more singular the intelligence.

Artificial General Intelligence

Although the preceding discussion renders the traditional notion of "artificial general intelligence" somewhat superficial, it is worth contemplating whether the term itself will bear any meaning if the AI systems—the artificial intelligence—is integrated with the current social intelligence.

While the terms "artificial", "biological" and "collective" intelligence distinguish the three kinds of architectures which exhibit agency and intelligence within their shared environment, we can easily recognize their selection as stemming from our basic intuition. As outlined before, these three examples are a mere handful from the myriad other kinds of intelligence that may be denominated. These are currently the most pertinent and their relationship is discussed and marketed. However, as different intelligent systems get integrated, as automation of human labor progresses, we may simply abandon the need to qualify the systems under the umbrella term AGI.

Today, we barely recognize companies and communities as intelligent systems, and "AGI" has a particular ring of foreignness that clearly juxtaposes against what we would ordinarily intuit as intelligence—other human beings. Nonetheless, when digital systems exhibiting agency begin permeating social structures, the distinction will likely vanish, as we will no longer perceive ourselves as distinct from our tools, much like we implicitly do with what has already been automated. We seldom refer to living humans as "human intelligence" or "human general intelligence", but rather opt to call them by their proper names—John or Jane—and, collectively, people. Instances of use of digital intelligent systems are more likely to be referred to individually—Siri, Alexa, Sydney, Bard and Claude sound much less artificial or otherworldly than the sterile "AI".

Prominent researchers in the field (e.g., (Levy and LeCun 2023)) have criticized the notion of "general" artificial intelligence, simply on the grounds of human intelligence's insufficient generality. However, the generality of

Universal Measures of Intelligence

Artificial General Intelligence

artificial intelligence seems even less of a meaningful denomination when intelligence itself is defined as the level of generality.

In fact, the generality of intelligence of digital systems will increase with the generality of human intelligence, as the two forms become coupled into a single hybrid paradigm. By engineering intelligent systems that exhibit agency in our world, we are reengineering our collective architecture and improving its generality and, hence, intelligence. In other words, the road towards “AGI” is simply a road to higher collective intelligence and higher social connectedness.

Simply by virtue of understanding how intelligence naturally evolves—intelligent systems organizing through communication into more intelligent systems, information being more concentrated within their confines—we may foresee a fast-paced informationally dense future society in which human beings are either partially assimilated into the technology or exist as a distinct form of a biological collective in symbiosis with the digital ecosystem, much like the microbiome of the human digestive system coexists with the host.

Language and Distributed Cognition

As we have seen before, the distinction between the agent and the environment is somewhat arbitrary, chosen at the intuitive boundary across which communication occurs. Our intuition leads us to presuppose that those parts of the system which are unchanging in their architecture and are mediating between two sides by allowing information exchange should be considered interfaces and are serving as a boundary between communicating agents.

Here, we arrive at a similar distinction: one between an architecture and a module. The same boundary with which we distinguish an agent from its environment in a complex interacting system is the one with which we may distinguish modules in the same complex system. If we suspend the term “agent” for a moment, and treat whatever is denominated with the term as simply an algorithmic component of the system interacting with the rest of the system, we see that an “agent” in that regard is nothing more than a mere component of the overall system to which we ascribe agency or a form of independence (or, even, a form of identity). The mere fact that we, as human engineers, cannot fully understand the inner workings of a module inclines us to designate such a module as having “intelligence” or “agency”.

In fact, the distinction is often both useful and necessary, as without separating individual modules within an architecture, it is impossible both to represent the architecture in order to understand its workings in the abstract and to exercise labor division if different components of the architecture need to be worked on by differently skilled workers. For example, in traditional software development, we are mandated to lay out detailed infrastructural and logical diagrams of our applicative solutions before beginning development and implementation, firstly in order to have a higher-level understanding of the whole system and to be able to plan development and, secondly, to assign appropriate skillsets and expertise for modular development. In other words, the modularity of the software architecture is, in part, a result of the modularity of the workforce.

In effect, a monolithic application that merges frontend and backend development is only possible if the engineers and developers building it are well versed in both frontend and backend development. Otherwise, a clear delineation of what is in whose bailiwick is necessary in order for the application to be built. Put differently, the skill disparity of the workers induces some of the modularity in architecture. To build a monolithic application of the kind I mention here one needs to have either a team of

Language and Distributed Cognition

developers who well integrate the knowledge of both frontend and backend development (i.e., are familiar with a full-stack paradigm or framework) or a team of diverse engineers who can communicate so well that the team acts effectively as a single full-stack engineer.

This is not to say that all architectural modularity stems from skill division amongst the workforce. Some architectural modularity is a result of technological shortcomings and the insufficiency of our understanding of how to build more tightly coupled systems which are energetically stable. For example, building large GPU clusters presents a distinct challenge in terms of unit intercommunication—the GPUs themselves are relatively tightly coupled internally in relation to other GPUs within the cluster. A major obstacle to overcome when building large computer centers is, in fact, GPU interconnectivity. Distributed algorithms, such as those necessary to train large language models, must accommodate the underlying architecture. Systems such as NVLink and NVSwitch were built for this purpose (NVidia 2014). Similarly, the computational bottleneck in systems distributed over a network is typically network interconnectivity. In other words, our lack of technological capability to connect individual components or machines into a single one is what causes the other aspect of architectural modularity. For example, if the communication interface between an edge device, such as a phone, and a cloud machine cluster were of sufficiently high bandwidth and sufficiently low latency, the entirety of client-server division would be rendered meaningless—the device and the machine cluster would simply be parts of the overall system, without a clear delineation. Without the communication constraint, the edge device would be borrowing its compute capacity, however miniscule in comparison to the machine cluster, to the overall system. If communication is so highly coupled that there is no loss (e.g., dissipation of energy on synchronization, error correcting, network routing etc.), then the edge device's compute capacity becomes an addition to the overall system's compute capacity, regardless of how small it is.

I argue, in fact, that the very reason why the industry is moving towards server-side computation is because we lack the means of effectively integrating edge into the compute, thereby resorting to making the edge simply an interface to the remotely integrated compute. Nonetheless, we can draw a clear conclusion: we intuitively view those parts of the system which are more coupled as single modules, while those parts whose coupling limits information exchange (i.e., where the coupling is loose) we perceive as interfaces or communication channels between modules.

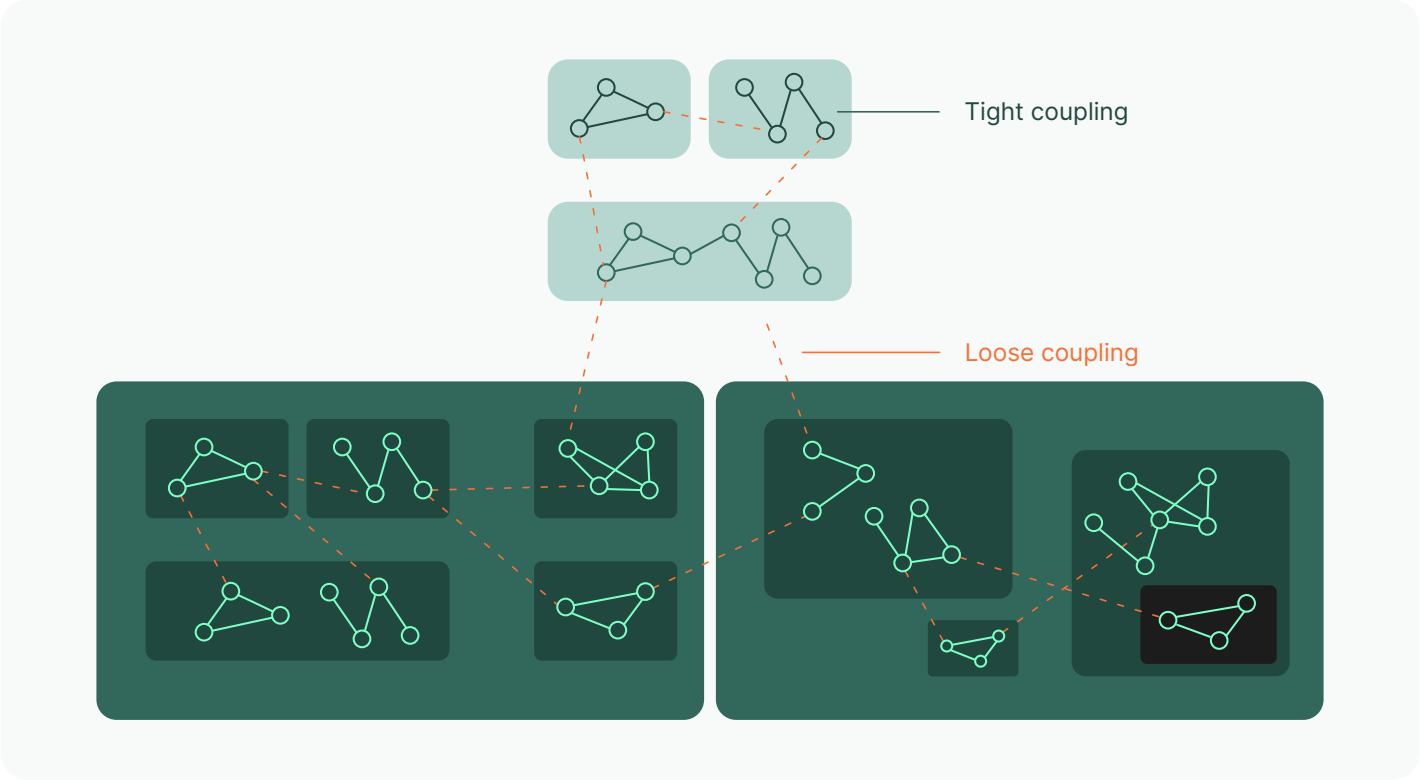


Figure 3. An interconnected system is intuitively divided into “modules” by the relative level of component coupling. Parts with tighter component coupling, which cannot be further subdivided, are perceived as modules, while loosely coupled connections are perceived as interfaces.

Language and Distributed Cognition

Thus, we must come back to the question of architecture. Any individual module may be “architecturalized” for the purpose of either understanding it or altering it in a meaningful way. However, the more coupled the components of a module, the more difficult it becomes to disentangle them from one another into a diagrammatic representation. Diagrams, and by the same token all architectural representations, rest upon the notion that there are modules which interoperate. If all modules communicate with all modules, the notion of an architecture becomes less useful. In fact, in these cases, an architect may be inclined to label the architecture as too complex or insufficiently modular and rearchitect the entire module. However, in doing so, the architect strips efficiency in favor of human readability, something that proponents of “clean code” advocate for primarily because code modularity allows for more efficient cooperation between diversely skilled developers. Modularity allows easier representation for a human developer, but comes at the cost of execution efficiency, performance and elegance. As long as humans are the main developers of code, this distinction will need to prevail to a large degree, simply due to the fact that most developers are not going to be polymathic.

However, industry experience has taught us that in scenarios where performance is crucial, it is necessary to give way to more traditional and monolithic approaches to software development. This is especially the case in embedded development, and, most notably, in operating system and driver development. Here, unit testing and testing in general must be

Language and Distributed Cognition

adapted to account for the integrative kinds of optimizations which are necessary in order to boost performance. Firmware-level optimizations often require reducing modularity to favor performance.

Similarly, in training large language models, the more general the architecture (i.e., the less modular and more monolithic) the more difficult it is to understand and represent. However, it has been argued that instead of attempting to impose a specific architecture when training general AI systems, we ought to simply use as general an architecture as possible with as many compute resources as possible (Sutton 2019). Although it makes sense to transfer some of our representations onto the model initially by way of biomimetic architecture of these systems, in the long-term, the more viable approach does seem to be generality, as, as we have previously discussed, intelligence itself is a measure of generality.

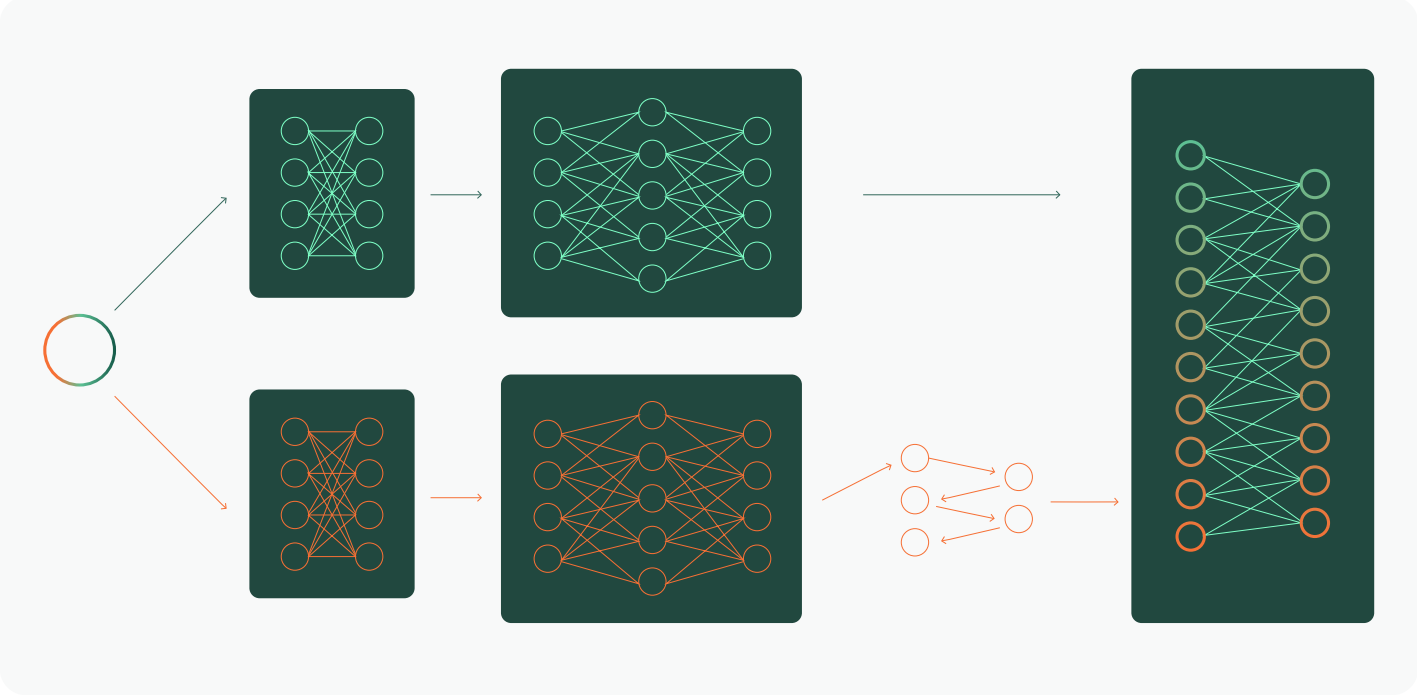


Figure 4. A complex AI system or neural network system comprises of multiple networks or modular components connected algorithmically. As systems generalize, components become integrated into a single indivisible module. What is interconnected is not modular. What is modular is not sufficiently interconnected.

The more singular the operation, the fewer the number of modules, and vice versa—modularity is a result of our inability to integrate components of a system into a single paradigm, be it because of our own lack of integrative understanding or simply a lack of an infrastructural foundation on which to build this integration.

Even a highly coupled module may be decomposed into its constituent elements. We would not call these elements “module” for the simple

Language and Distributed Cognition

reason of there being too many and modularity, in a way, implies simplicity. Nonetheless, for any system there exist interacting components and their interaction is either perceived as *computation* or *communication*, depending on the level of their coupling. We say that modules communicate, but we rarely consider their processing as internal component communication. In other words, if the components are sufficiently highly coupled that clear lines of information exchange cannot be determined by observation, we call this information exchange computation, rather than communication.

However, the fact of the matter is that in both cases information exchange occurs and the distinction between communication and computation is one we make arbitrarily and, arguably, the same arbitrariness that guides our distinction of an agent to environment, or module to architecture, is the one that guides this delineation.

If our goal is to build systems which exhibit higher intelligence, we must dispense with the notion of modularization and build more general self-modifying architectures. Any static architecture is going to impede the goal of generality. The more general a system, the less modular it becomes. Effectively, to engineer a highly intelligent system, we must aim to engineer communication in such a way to allow it to become the facilitator of computation.

In fact, the same argument applies no matter the *kind* of intelligence in question—engineering communication to facilitate distributed computation ought to be done in corporate, biological and technological systems, if the goal is maximizing intelligence.

In a corporate or communal setting, an individual who is either not sufficiently general (i.e., polymathic) in their knowledge or is inadequately connected to the corporate or communal core will be naturally excluded from the core computation process of the corporation or community in question. In more metaphorical terms, the same relinquishment of computational ownership that happens with edge devices in favor of the more coupled cloud cluster will manifest as worker/citizen enfeeblement with regards to the community core—the specialized worker is becoming the company's edge device. Fundamentally, if the worker is not general enough in their skillset and adaptability (i.e., in their intelligence, according to the definition I laid out above) and if the bandwidth of their communication with the company core process is limited, they are likely to be entirely excluded from the process. The conclusion here is that unless the worker can offer general enough intellectual compute resources which integrate well with the company, they will be excluded in favor of a more suitable one, be that a human being or a digital agent.

Note that I am making neither a technological nor a corporate argument—it is not the question of whether less intelligent workers will lose their jobs or whether companies ought to do this or that—but simply an argument about intelligence in general. It has always been the case that workers were selected into the workforce based on intelligence (Herrnstein and Murray 1996) (Deary 2020), however, the argument I am making is that

Language and Distributed Cognition

this is the natural behavior of intelligent systems, in general: increasing intelligence implies increasing the coupling and generality of the system, in turn reducing its modularity and component specialization. In fact, the same faith awaits both the low-intelligence and high-intelligence workers if the core intelligence of the corporation exceeds their own—they are gradually going to be relegated to the edge.

What is more salient to this discussion is the notion that human-to-human communication is itself a form of distributed computation. We are coalescing through communication to jointly solve a set of tasks at hand. However, as we engineer digital systems whose intelligent components can communicate more efficiently than we can through our language, unless we find ways of upgrading our linguistic capabilities, we are going to be left behind. It will likely be innovations in user experience design which allow us to keep participating in the core computational processes of the communities and companies we are part of.

The same argument may be applied at multiple scales: individuals not coupled to organizations will gradually become modularized towards the edge, mid-sized companies insufficiently grouped will be pushed to edge, as well as communities, societies, countries—increasing the intelligence of the Earth (i.e., biology, society and technology combined) inevitably leads to gradual exclusion of all entities (be it legal or biological) which cannot be efficiently integrated into the core process. This gradual exclusion is going to prefer less general and less connected over those which exhibit higher adaptability. In other words, growing intelligence necessitates preferring integrating higher intelligence at the expense of lower. In effect, as intelligence is built, modules become less prevalent towards the process center, where information is more integrated and more prevalent towards the edges where greater specialization occurs due to lack of generality.

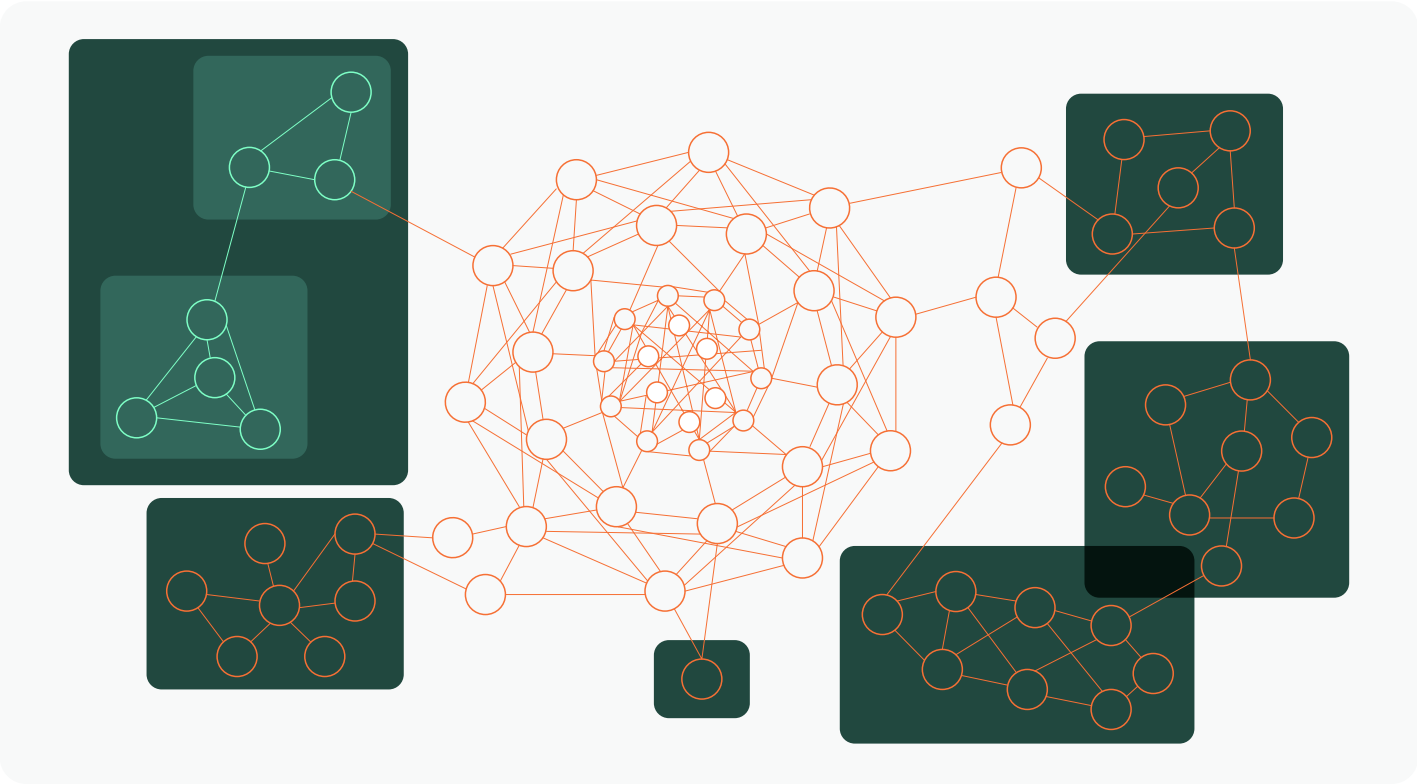


Figure 5. An intelligent system exhibits less modularity in its core and more modularity towards the edge. Modules are only perceived where there is insufficient connectivity to entangle with the high-entropy interconnected core of the system.

Language and Distributed Cognition

In order to facilitate human inclusion in the workforce while scaling intelligence, we must account for the insufficiencies of our natural communication systems. Currently, we are limited to the means of communication nature has bestowed upon us, namely our verbal and body language. Through current generation technology, we are able to extend our modes of communication to typing and, in certain scenarios, VR and AR experiences, using digital twinning and visualization. However, the bandwidth of these remains vastly inferior to that which is required for more complex communication and although there are early attempts of mitigating these, such as Neuralink (Elon Musk 2019), they are in early prototyping stages.

Nonetheless, the emergence of neural interfaces is indicative of the very trend indicated by our discussion thus far. The high degree of connectivity modern society is exhibiting is a symptom of its growing information integration and, consequently, intelligence and our integration within that system necessitates changing our interface with it. In effect, we are attempting to find ways to bridge our natural communication bottlenecks and obstacles, but in doing so we are relinquishing individuality—our personal distinctions as “social modules”, being independent autonomous individuals participating in society of our own volition—in favor of the collective form of intelligence.

Language and Cistributed Cognition

As said before, this discussion is not to serve as an argument for or against such progress, but merely to elucidate an inevitable outcome, should our progress in developing higher intelligence continue. Simply put, we cannot remain “in control” of society without being part of it and thus the future is either one in which we are collectively gravitating towards a more singular intelligent entity or are simply residing on a platform which, through its superior intelligence, which we hold no understanding of or control over, is allowing us to exist. If we wish to retain our individual sovereignty, then we must accept the latter outcome (i.e., relinquish individual understanding how our society works). However, if the goal is maximizing our understanding of society and, presumably, nature, then we must relinquish individuality in favor of the collective.

In fact, this relinquishing of individuality has been happening ever since the inception of technology and society. Society is what enables distributed representations to exist. Language cannot fulfill its purpose if its symbols do not stand for, at the very least, similar concepts to those using it to communicate. The nature of language is such that it allows us to exchange pieces of our mental symbols in order to distribute our own meaning across the collective, as well as uptake others’ meaning into our own mental representation of the world. In such sense, language is the means by which we execute our collective computational process—we are the coupled components in which collective representations reside. The more sophisticated the language, the more coupled its systems, the more distributed the collective representation.

Our experience of shared understanding and values is, in a sense, a direct reflection of representational superposition (Elhage, et al. 2022) (Henighan, et al. 2023) which occurs when a concept is distributed across a system. Our interpersonal communication facilitates collective computation. Individually, we are exchanging symbols in order to better understand the world around us, and, consequently, update our world-model and concepts, but, collectively, we are facilitating a kind of computation that has the same aim as any individual—updating the collective world-model to preserve and advance the collective into the future.

Agent Communication and Interface Negotiation

A large language model based on the transformer architecture is essentially a next-token prediction machine. However, LLM viability in practical scenarios, the main promise of artificial intelligence, is contingent upon the possibility of expanding mere text continuation to other applications, the most known of which is conversation. Fine-tuning LLMs for turn-based conversation—conversation in which pieces of text are attributed to roles (typically, the system, *assistant* and *user* roles, for setting conversation parameters, marking assistant responses and user queries, respectively)—is the first and arguably the most intuitive route to take. In fact, most of the LLMs released in 2023 and 2024, have been fine-tuned and adapted for turn-based chat conversation.

This role-based turn-based setup provides a platform for other types of interaction and functionality, as the models can be deliberately tuned not only for instruction following, but for role-based rule obedience. One such evolution was the introduction of the *function* role in, for example, the OpenAI API. This kind of specialized message role informs the model of the kind of use the content of such a message is meant for. In other words, system tokens are used to mark the type of message, and this message type guides the model's response and interpretation to the message's payload. Clearly, this approach allows for a kind of integrated quasi-authorization or instruction-level privilege. Concretely and as an example, a message marked with the *function* role allows the model to better understand the intent of the user, and, more importantly, the intent of the system. Here, the "system" implies all those automated algorithmic components which route messages between models, instruct models automatically or from predefined templates, log, or parse model outputs.

Taking this approach a step further, an engineer integrating LLMs into an agentic or semi-agentic system meant for digital robotic automation can now fine-tune message types to align with the different aspects of the concrete system being built. More concretely, we are now able to fine-tune for new message types, appropriate for the specific system we are developing. For example, if we are working on a multi-agent system where differently configured agents (with different roles and system prompts) are set up to interact with one another conversationally, it is beneficial to specialize *user* messages by fine-tuning additional specifier tokens. In such a context, each agent represents a user to all others, but the user identity is embedded into the user message specialization. That way, instead of a single *user* role, the system might use *user-0*, *user-1*, etc., to implicitly identify agents to

Agent Communication and Interface Negotiation

one another. This approach saves on tokens by removing information that would otherwise be provided through the *system* message or as a header to the relevant *user* message.

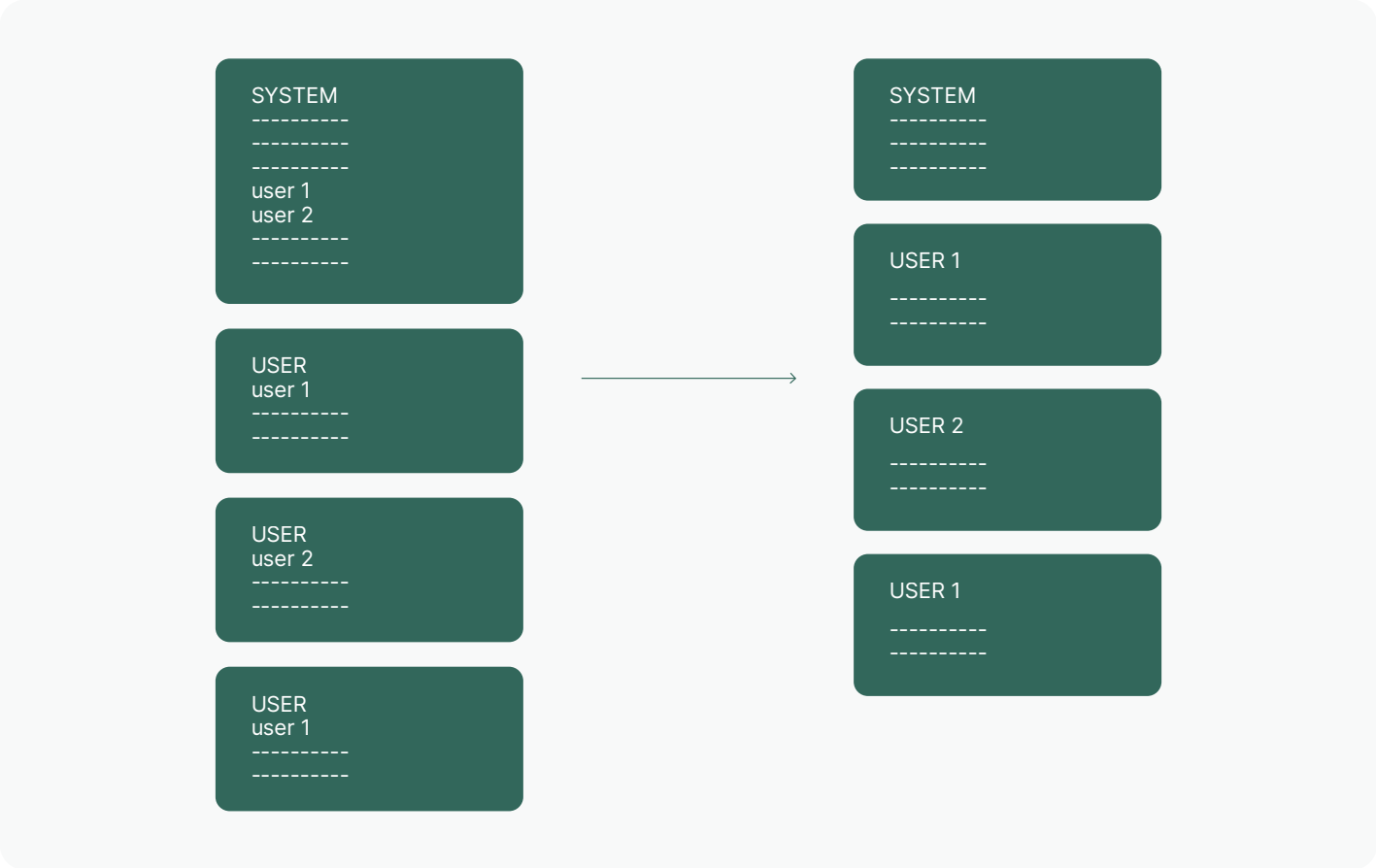


Figure 6. User message headers can be compressed into user-message tokens through fine-tuning, reducing the token count.

However, this approach is rather rudimentary and somewhat of a hack. A system where the transformer-based text-completion model is adapted not for conversation but for specific function-calling may be a much more viable option. Similarly to how OpenAI's o1 family of models was trained to produce different kinds of tokens (in that case, so-called "reasoning" tokens), or how Anthropic fine-tuned their model to enclose reasoning or fragments into html-like tags, one might expect the next iteration of interaction-enabled models to be fine-tuned precisely for that purpose.

As we have seen with Anthropic and others, XML is increasingly becoming the de facto standard for specifying intent—so much so in fact that one might question the need for message roles. If a model is trained to recognize the semantics of an XML section based on the attributes specified (e.g., specifying `<message role="user-x">` instead of using a `<|user|><|user-x|>`

Agent Communication and Interface Negotiation

token combination), we can dispatch with the message-based approach entirely. In that case, the modality and semantics of an XML section can be expanded well beyond just user roles.

Coincidentally, XML is predominantly used in its more widely recognized form in client-side user-interface code, as HTML. In fact, models trained to produce meaningful XML implicitly produce semantically infused user-interface code. Each piece of properly formatted XML produced by a model trained in this way is, in some broad sense, a user-interface prototype. Here, user-interface is meant more generally: a user-interface between two agents is any semantically structured piece of text that can be effectively and reliably recognized by both sides in the interaction. In other words, simply designating the source agent's identifier through an XML attribute will inform the other side about the content's originator and their intent. Note that this works regardless of whether the source agent is an LLM or a human.

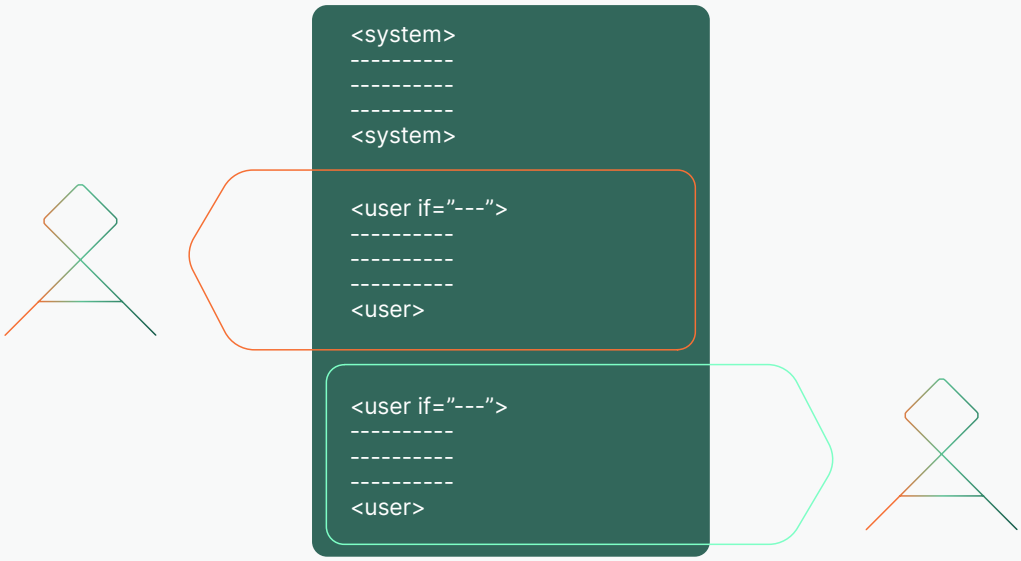


Figure 7. Agents view messages intended for them or sent by them. They interpret XML tag text as message container boundaries.

This way, instead of the model being trained for mere text continuation, which will always remain a fundamental aspect of transformer's operation, it can be built for the purpose of DOM parsing and manipulation. Such a model has the same semantic potential while opening novel communication possibilities. In other words, the context window is now, by design, meant to be used both as the means of temporary data storage, reasoning and

Agent Communication and Interface Negotiation

communication, all of which is properly semantically structured, linked and differentiated. This kind of thinking about the context window allows certain pieces of text to be shared across agents: two agents having read-only access to shared XML sections, but only able to write to sections they own, while the system which governs their interaction manages permissions.

Systems trained in this way would implicitly move away from simple agent interaction to a more collaborative interaction model. Because the DOM is partially owned by different agents, a degree of concurrency in communication is allowed by design, whereby different agents can lock DOM elements and write and read from them.

Most importantly, however, the fact that the shared piece of text is formatted as XML and imbued with semantics allows the text to be rendered into a different form, namely that of a user interface. A human agent easily plugs into such a system through a browser-like interface which renders the XML according to a predefined schema into UI components that the human agent/user can now interact with. What results from this approach is a new kind of interface design: a negotiated interface.

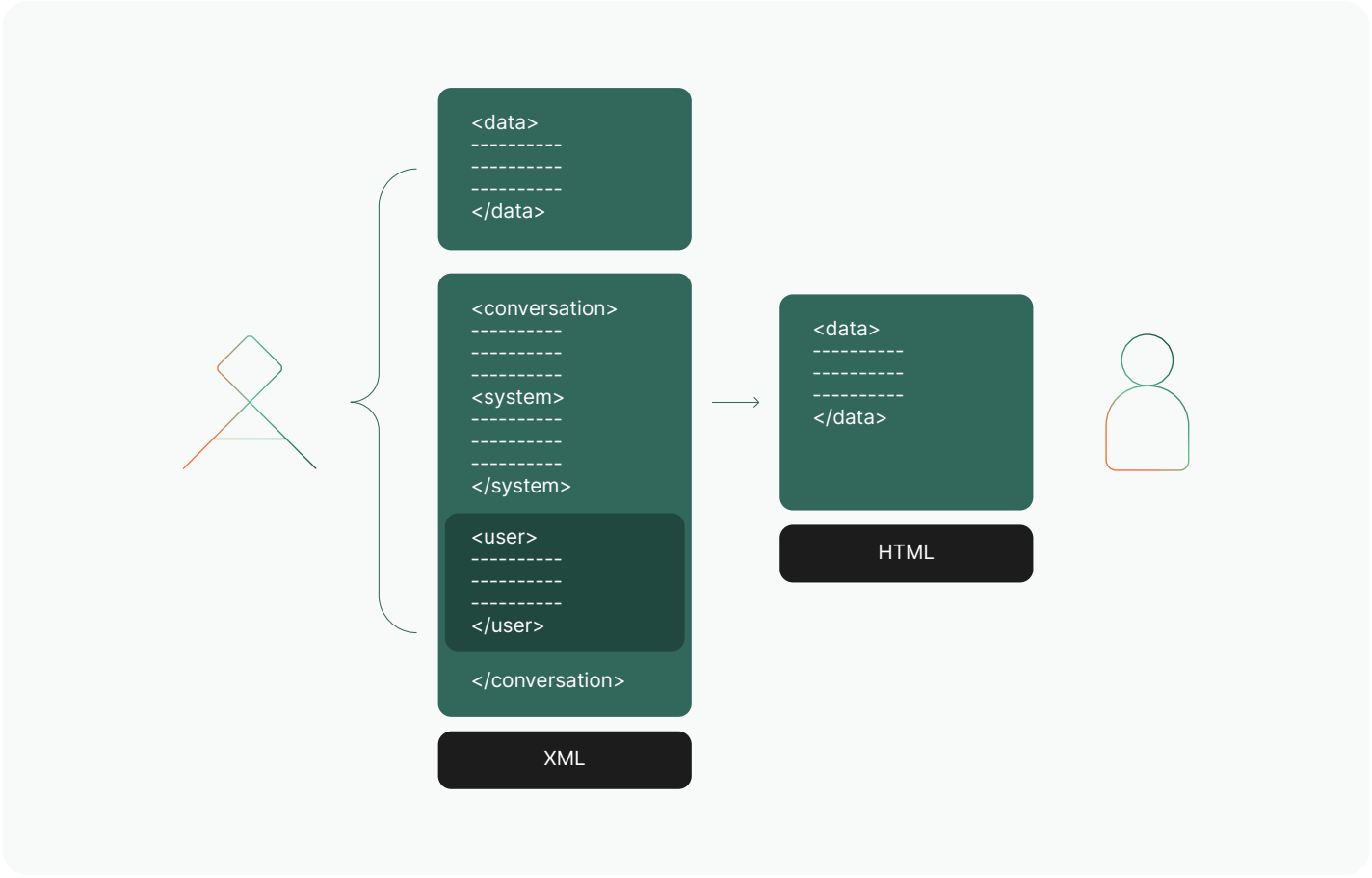


Figure 8. XML elements are easily reinterpreted and rendered as HTML elements representing UI.

Agent Communication and Interface Negotiation

Given the development direction most AI providers are taking, this kind of negotiated interface design is likely the next step. In this future, the role of the frontend engineer is going to become significantly more abstract: the engineer will be the one specifying the parameters of the agent negotiation—the negotiation contract—while the participating agents themselves will design the interface through the mere act of interaction. The UI, in this scenario, changes in accordance with the flow of communication, which no longer implies simple conversation, but a multi-modal interaction between two or more agents. Each agent is prompting the other both with the goal of solving the problem at hand and with the goal of optimizing the communication mechanism itself.

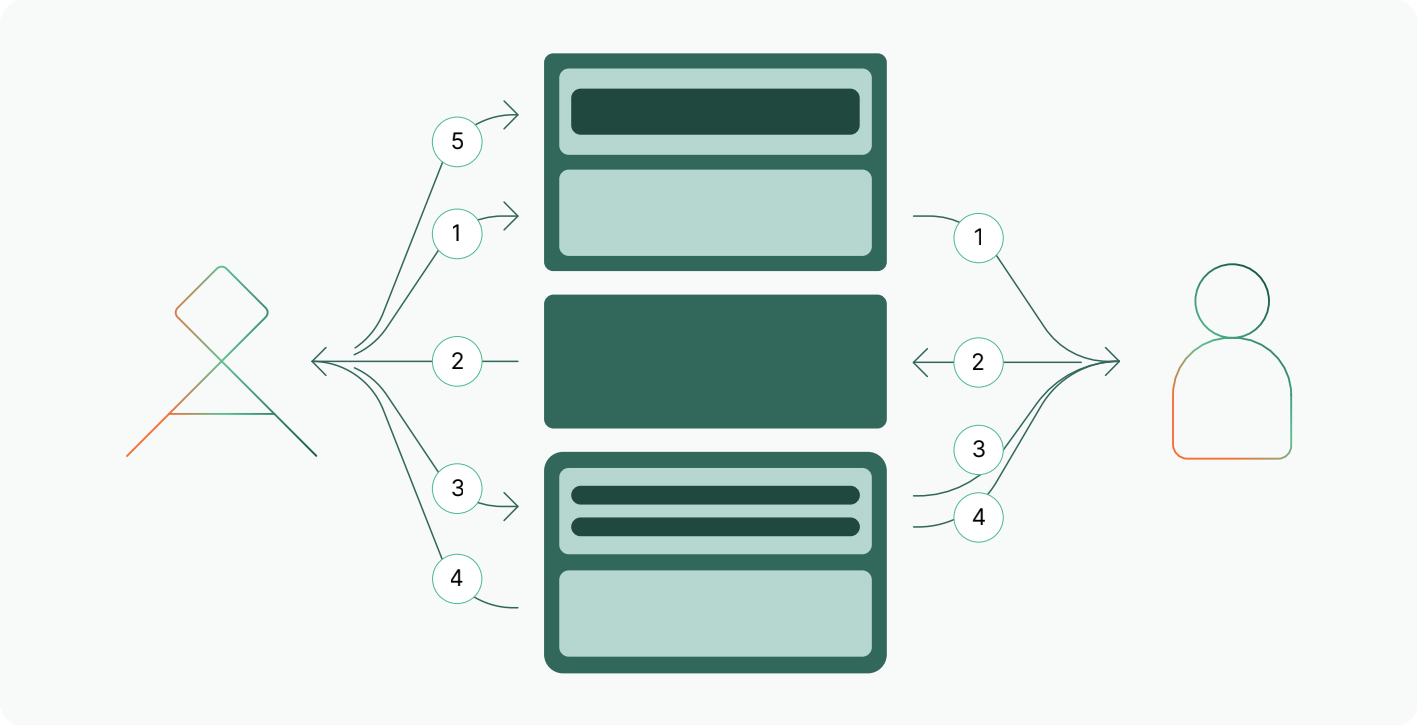


Figure 9. Message exchange through XML may be done by insertions and reads from a shared XML file.

More pictorially, instead of asking for a clarification or a description from the user, an agent may simply create the appropriate UI—include a color-picker, a slider, a rich text box, a set of buttons etc.—most relevant to the current conversation and then, when the nature of the topic changes, adapt, remove elements and alter the UI, so as to keep it minimally cluttered, but maximally effective. In a sense, the agent becomes a real-time on-demand interface designer.

Note that such a communication style would emerge from the model being trained to output semantically imbued XML, rather than being trained for turn-based conversation. Models making use of that mode of operation harbor a much greater semantic potential and are much more general in their use: while they may be applied to simple chatbots, they can also be

Agent Communication and Interface Negotiation

leveraged for reasoning, data transformation, data retrieval, function calling and tool use, and distributed agentic computation, without demanding hacks and circumventions of the provider's intention—models designed this way are engineered to be more general agents, rather than being meant simply for conversation and then adapted for other use. In other words, structured text manipulation is a more general intelligence-building capability than chat, while providing chat functionality as one of its specific use-cases.

Instead of building models specifically for communication, it can be easily argued that to get to a more general form of intelligence, we must build models from whose design communication emerges as one of the modes of operation and use.

Although language facilitates communication, more broadly and more technically, it is a tool for computation (Sipser 1996) (Hopcroft 2008). In fact, any consortium of human agents may use language in broadly two different ways: one, for consolidating the collective representation (sharing information and ideas) and, two, for solving a given problem collaboratively. In that sense, a system with general intelligence ought to be able to reliably switch between those two kinds of operation.

In a typical work environment, humans coordinate to either optimize the social structure they are part of by means of changing, assigning, and delegating roles, whether they be management or execution, and to distribute work amongst themselves based on aptitudes and prior experience. To foster that kind of approach to collaboration, we rely on practical tools—software, equipment, materials, paraphernalia—we switch between different environments, we change our perspectives and team member structure to facilitate the natural flow of the conversation towards a solution. In other words, we are, without giving it much thought, changing the conversational context (be it by inviting someone else, deciding to use a whiteboard and a marker, or opening up a coding environment), so that we can more easily express ourselves or understand the other side—we are, *ad hoc*, negotiating our communication interface as we carry on with the conversation.

In order for a digital agent to truly integrate into a real-life ecosystem, it must be able, at least to a degree applicable to software, be able to support such a natural alteration of the communication context. A generally intelligent model ought to be more general than chat and be able to select chat as appropriate mode of operation, when that mode is warranted by the current context.

The experiments conducted in relation to this treatise indicate that current generation models, such as o1, GPT-4o, and Claude Sonnet 3.5 are already adept at recognizing and amending semantics of an existing piece of text structured as XML. Informing the model through the system message about the semantics of predefined tags and attributes yields relatively reliable results. Fine-tuning for this purpose, based on produced sequences with highest rates of success will, by design, yield better results. However, as outlined before, this is a hack requiring a kind of reversal of the initial intent.

Agent Communication and Interface Negotiation

Although it is a slight speculation, we can expect the larger AI service providers aiming for the ever-elusive AGI to take on a variant of this training approach at least in some of their models, and so it stands to reason that we ought to prepare our agentic systems to integrate with that paradigm. Even if the prediction turns out to be incorrect, the existing turn-based conversational models can demonstrably be adapted, albeit with a degree of dirty work, to perform quite successfully in the more general context.

The main issue in this context remains latency. Even when implementing partial asynchronous rendering of the DOM elements being regenerated or edited, the user must still wait for the generation to end before the element can be refreshed, given the token generation speed of the large foundational models, the approach is still infeasible in many practical scenarios. Nonetheless, creating simpler systems which make use of the XML generation to enable adaptive interfaces is very much within the realm of possible.

The future in which your Teams, Slack or Discord bot adapts the UI from chat to chess is still quite a few years away, it is nonetheless a very promising prospect looming somewhere amid the mist of possibility and it is well worth preparing for, if anything, as a side-effect of investigating alternative model capabilities.

Intelligent Operating Systems

Fine-tuning, as we have seen, for more general purposes than simple turn-based conversation, such as for XML-based general interface negotiation, provides a more fertile ground from which to build more complex agentic systems. Today, while existing models such as GPT-4o, Lamma 3.2 or Claude 3.5 are showing promise when fine-tuned and/or prompted for such behavior, the execution of such systems remains a technical hack, given that the model needs to be detrained from its biases towards existing system tokens and, consequently, text segmentation and structuring. Nonetheless, we can see significant leaps in agentic behavior through the use of more general XML-based structuring.

Here, I will endeavor to explore one such approach in a virtual operating system environment. For the purposes of this experiment, I designed a specific XML-schema and mode of operation, foremost to illustrate that such agentic functionality is feasible in practice. I will outline the approach and architecture, and provide the rationale behind it, but it nonetheless remains firmly within the bounds of research, due to the fact that the models are not yet intended for such use and, as we will see, fail in certain key scenarios. Nonetheless, our experiments clearly indicate that truly agentic systems would benefit from such a micro-paradigmatic change.

Our example will include exact details of how a current generation large language model might be used to augment operating systems agentically and imbue them with a form of intelligence proper. For this reason, I must assume that the reader is well acquainted with the fundamentals of operating system design, namely how programs and processes are segmented, what privilege levels exist and how code is run on the physical machine (Tanenbaum and Bos 2014) (Silberschatz, Galvin and Gagne 2021).

I defined a minimal set of necessary components for a toy intelligent operating system on which to explore the behavior of current generation large language models. Although the toy system showed great promise, the aim was not to attempt a full-scale implementation, as it would be costly and require training from scratch which is something that I envision as the inevitable future evolution of agentic automation and AI integration. For this purpose, I should emphasize that the presentation below is not for the purpose of advertising this specific approach, but rather for showing how such a system might be constructed, informed by my own and my teams' experiments with the implemented toy system. I encourage the reader to treat the following presentation as an illustration of how future operating systems might implement intelligence. My intention is to

explore the avenues of building highly intelligent systems from smaller components and integrating artificial intelligence with human intelligence. As we will see, fundamental paradigmatic changes in how we think about the relationships between programs and processes, users and programs, and even source control and memory management will be alluded to simply by virtue of exploring a simple toy model of an intelligent operating system.

In order to design an operating system that permits true agency, we had to re-envision how development might be done in the future and what systems need be used to facilitate software development, management and work in general.

The Structure of an Agentic Program

The essential conceptual change between the notion of a traditional user-space program and an agentic program is that the program's code is treated polymorphically by multiple different virtual machines. Since the prototypes used to inform this text were created as virtual machines executing in user-space, an agentic program prototype's definition was implemented simply as a programming language extension of a well-known programming language (tests included C# and JavaScript, depending on the prototype, but the choice of language for the proof of concept was essentially arbitrary). In other words, the implementation was written not to run directly on hardware, but in a user-space virtual machine written in a higher-level programming language, where the different agentic program segments were executed and routed for execution by components written in that language of choice.

In a more metaphorical sense, we are attempting to imbue highly predictable deterministic programming code with non-deterministic intelligent behavior by "executing" natural language instructions as part of the code, effectively switching between deterministic and natural language modes.

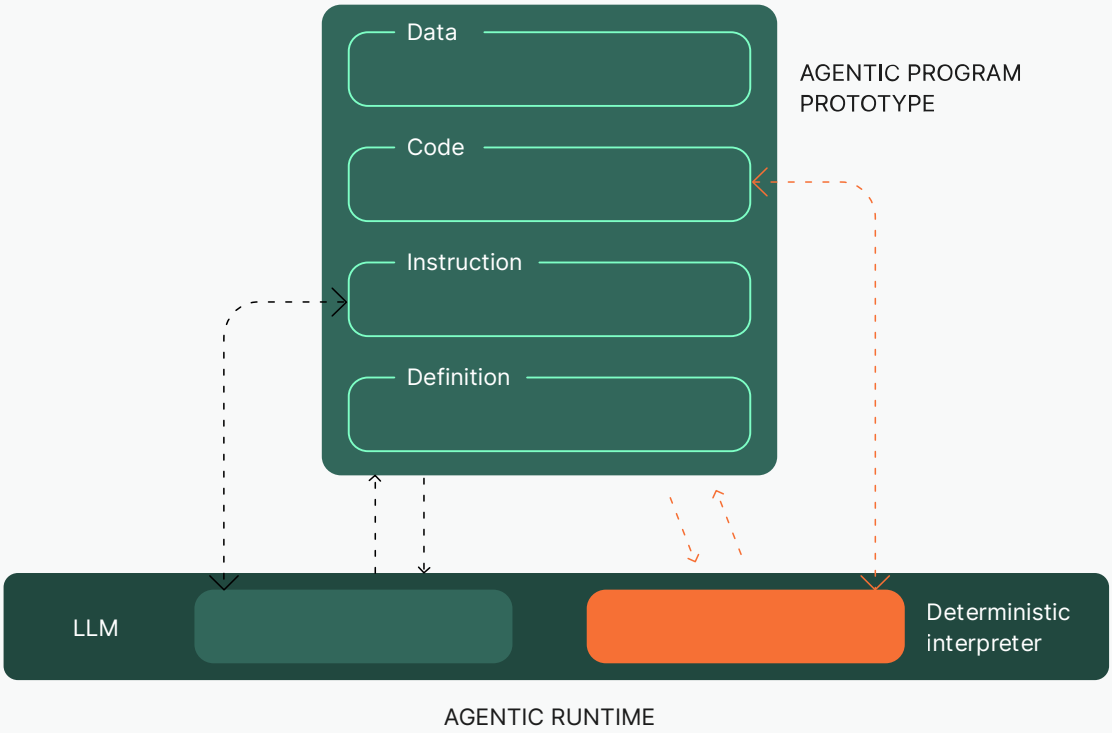


Figure 10. Agentic Runtime consists of a deterministic interpreter and a natural language interpreter (LLM) coordinated algorithmically for context switching.

Intelligent Operating Systems

The Structure of an Agentic Program

Practically, this means that a program prototype is essentially a piece of text interpreted simultaneously by two major virtual machines: one, a traditional deterministic runtime, such as a containerized subset of the .NET runtime, and two, an LLM-based non-deterministic runtime. Here, the well-known XML structure becomes highly appropriate. In a similar fashion how one might use XML as a basis prompting mechanism to both elicit turn-based conversational mode to manifest and to allow for user-interface rendering based on the same prompt, we may use XML to specify a prototype for execution that is segmented, through the use of XML-specified DOM, into segments interpretable by both a deterministic and a non-deterministic interpreter. This way, the very body of the program prototype contains tags which encapsulate what would traditionally constitute a program's memory layout: the code segment, data segments, heap and stack. However, the program segmentation is made more versatile in this way, in that it allows for future extension as well as routing to multiple underlying interpreters. A trivial example would be the one in which a piece of C# code may be interpreted, step by step, by an LLM (a case that arguably isn't the best use of token inference time), or, simply, executed by the JIT compiler of the C# language's native runtime.

The example given above is trivial in that it does not bring much practical value. However, the converse case, in which a prompt segment—one written in English, French or, say, Ithkuil— is loaded into memory and executed by

Intelligent Operating Systems

The Structure of an Agentic Program

an LLM is unique to this form of agentic programs. Where the merits of this approach really shine is in the interplay between the two underlying interpreters. When a prototype—our agentic equivalent of a tradition program—is loaded into the virtual memory, it becomes an active structure, much like a traditional process and its components in memory may change by virtue of its different executable segments being executed by either of the runtimes.

For example, a prompt segment written in aforementioned Ithkuil may, after a given length of time allotted to chain-of-thought reasoning, produce a series of system API calls issued to the underlying virtual machine. Those system calls may include appending items to the existing DOM, or changes of existing segments, including code segments. More importantly, the fact that XML elements can be named and otherwise decorated through attributes allows defining entry-point functions for each segment, including prompt segments. That way, a traditional C# code segment may invoke a Claude 3.5 prompt segment by issuing a call to a predefined system API method, which, in turn, may invoke a method written in another segment in JavaScript and so on until the entire starting segment is executed.

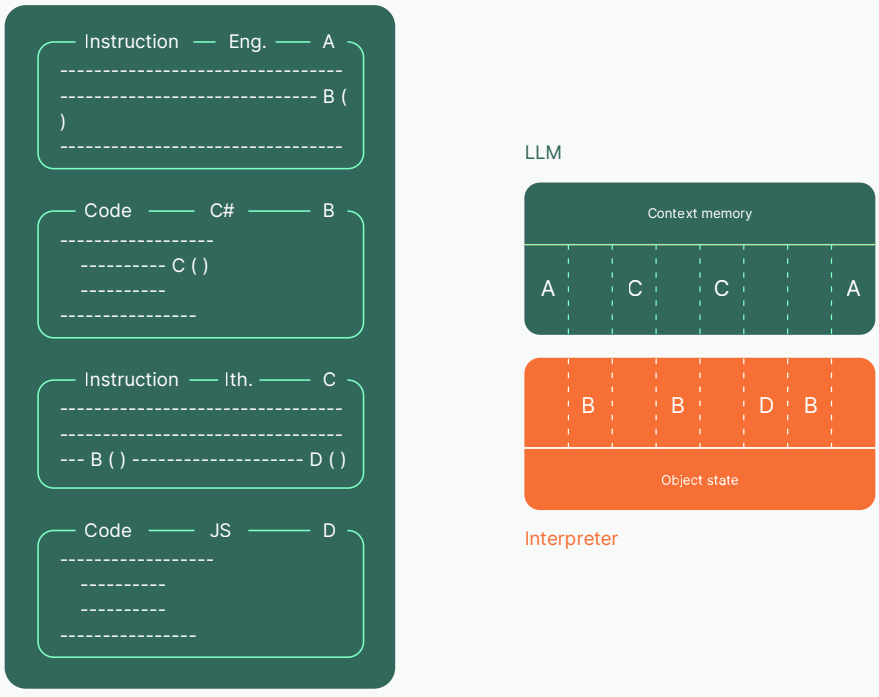


Figure 11. An example of polymorphic method calls (natural language from code and vice versa) and the corresponding execution flow and context switching.

Intelligent Operating Systems

The Structure of an Agentic Program

This way, an element of a simple prototype has at least three polymorphic interpretations: first, as a piece of deterministic code which may be executed by the runtime designated through its attributes (say, TypeScript or Python), second, as a textual prompt executed by an LLM designated through appropriate attributes, and third, as a hyper-text specification of a user-interface to be rendered to a secondary agent sharing the prototype (say, a human user interpreting the prototype with a prototype browser).

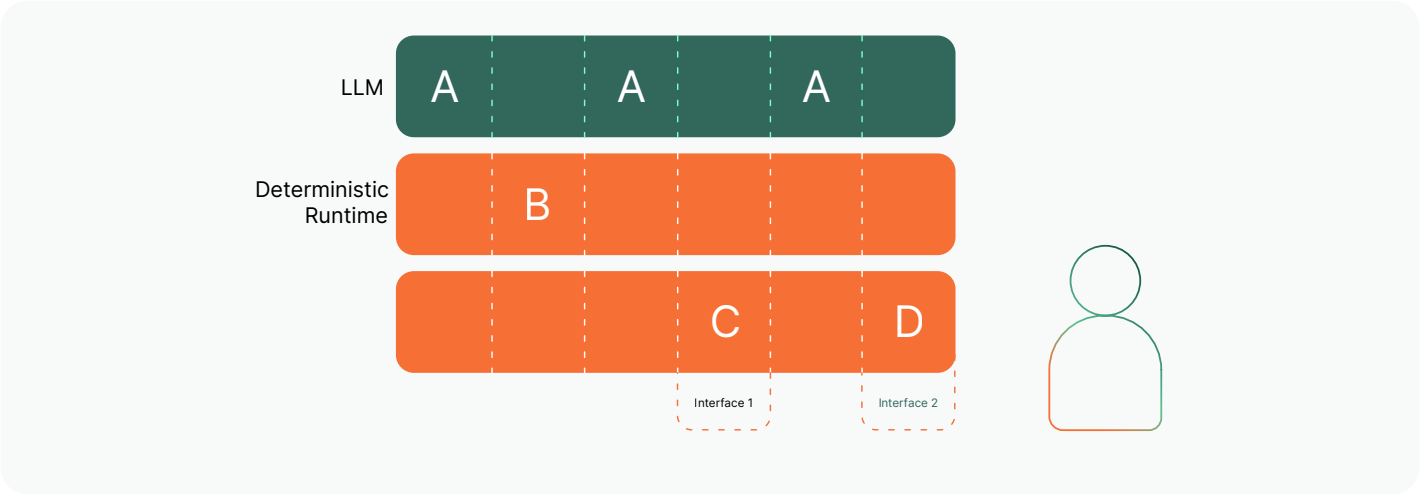


Figure 12. Humans may be included as a third kind of interpreter within the Agentic Runtime—they operate in a similar non-deterministic fashion as LLMs, but interact with the Agentic Runtime through user interfaces, implemented as software adapters or applicative software.

The underlying virtual machine, or, if implemented on metal, the operating system, essentially provides a standard system API which can be invoked in different ways by different segment types—for deterministic code segments, simply by making a call to a predefined method, much like it would be done with any standard library from high-level code, and for non-deterministic prompt segments by means of outputting a specifically decorated XML element (in myt experiments, I used the `execute="true"` attribute to designate any segment that should be immediately executed by the constructed virtual machine). The virtual machine, or the *agentic runtime*, is monitoring the outputs of all encapsulated LLMs, as they generate tokens. As soon as a parsable structure is output, it is routed to the appropriate interpreter and executed.

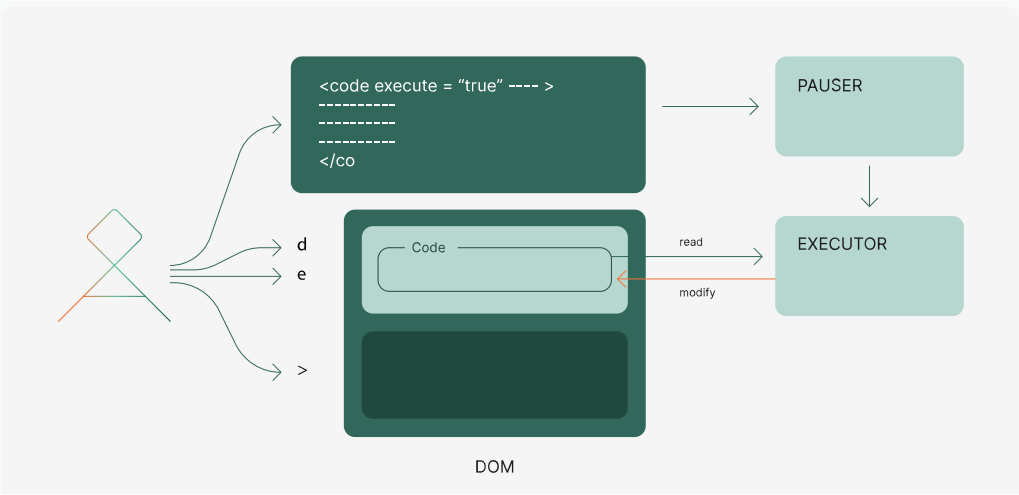


Figure 13. Parser component awaits a correct output from the agent and forwards a correct match to the executor which then performs the relevant modification, insertion, deletion or read operations from the global prototype memory and its document object model (DOM).

Intelligent Operating Systems

The Structure of an Agentic Program

Finally, we must briefly turn to the memory model of such a system. Given that an agentic program prototype is nothing more than a piece of plane text residing in the virtualized working memory, we cannot speak of traditional memory locations or memory indexing, but rather only about XML element access and retrieval. Obviously, even in an operating system built for this purpose, the agentic program would be nested within a traditional program in some way, simply due to having to accommodate for the Von Neumann architecture. However, we must treat an agentic process as if it is operating against multiple types of memory, as we are relying on multiple interpreters (namely an LLM and a traditional runtime) simultaneously. Given that a prototype may be larger than a given LLM’s context window, element attributes are, in our approach, used to specify the kind of virtual memory allocation (as well as access privileges) for the element. This way, the agentic runtime creates a view of the running agentic process which is supplied to a given LLM’s context. This way, the agentic process is running in the standard working memory, but the chosen view itself represents the currently running LLM’s working memory. Additionally, the runtime may contract certain XML elements and allow the model to invoke a system call to reveal their contents (this was our approach to RAG in this agentic scenario). From the perspective of the running LLM, there is no distinction between persistent storage and RAM—both cases are handled by the agentic runtime and processed through LLM-facing RAG API—but code segments executing on the traditional runtime have distinct access to either persistent storage or traditional working memory (in effect, they are traditional interpreted programs). Beyond this, the memory model is further extended for the specific approach to implementation used, which is of less importance for the general discussion here.

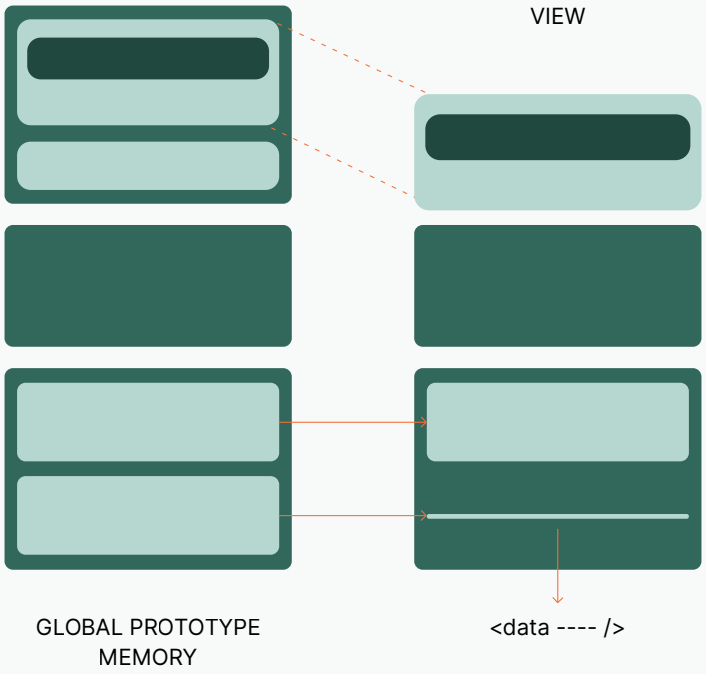


Figure 14. Agentic Runtime creates the appropriate views for each agent connected to the global prototype memory. Agents only read and write to segments of the memory made available to them. This reduces agent context load and allows for user authorization.

Clearly, there are many hurdles to overcome when constructing such a system, some of which include exception handling, semantic errors, hallucinations/confabulations, concurrency, memory management, storage persistence and many others. However, we will focus on a single implementation and attempt to address the most meaningful and impactful aspects, with the intent of exploring the possibility of engineering distributed intelligent systems that manifest higher forms of intelligence.

Example Prototype Structure

For the exemplified approach, I aimed to define a minimal set of element categories which would make it possible for the entire intelligent operating system prototype to operate and exhibit higher-order intelligent behavior.

I specifically defined six main section types which may be nested within one another: data, instruction, code, process, definition and task. Each section, as discussed previously, is represented as the eponymous XML element enclosed by the corresponding XML tags. We will enumerate the element names along with the most important attributes.

Data sections

All information available to LLMs, as well as information available for retrieval through deterministic code execution is enclosed within data elements. Essentially, any information provided within this element is treated by the executor (agentic runtime component, such as an LLM or a deterministic interpreter) by the types specified through the `data-type` and `data-adapter` attributes of the data element. To avoid unnecessarily detailing the attributes, it should suffice to say that the two mentioned attributes inform the agentic runtime about the way in which it should treat, reformat, or transform the data contained within the data element. Fundamentally, the contents of a data element are merely a textual representation of some underlying blob stored somewhere in memory and made available to the executor in this format.

As mentioned before, the agentic runtime keeps track of all the references pertaining to stored data—not only documents, images, and audio, stored on the file system, but also relevant in-memory data structures, objects, variables and even stack state of executing code snippets—and presents them to the executor (e.g., an LLM holding a view over the agentic process or a script executing against the current context) as the data element.

Given the polymorphic nature of the agentic prototype, when a human user interacts with the prototype, they are presented with each element in a way which corresponds to their native way of viewing information. For example, if a data element was present for which the data-adapter attribute was set to `data-adapter="audio"`, a human user (human executor) would see a rendering of an HTML audio element and the agentic runtime will have provided the necessary transient link for hearing the audio. However, were the user/executor an LLM, the corresponding adaptor would have rendered the audio file into a textual representation, based on adapter's implementation (in fact, the adapter itself need not be written in, say, C#, but simply be another agentic program). Furthermore, were the user/executor an LLM capable of receiving and interpreting an audio file at the input, the agentic runtime would have formatted the model request JSON accordingly, by supplying the actual audio file. In other words, the set of data adapters is specific to the executor and information is presented within the executor's view based on their set of adapters, regardless of whether the executor is an agent or a human.

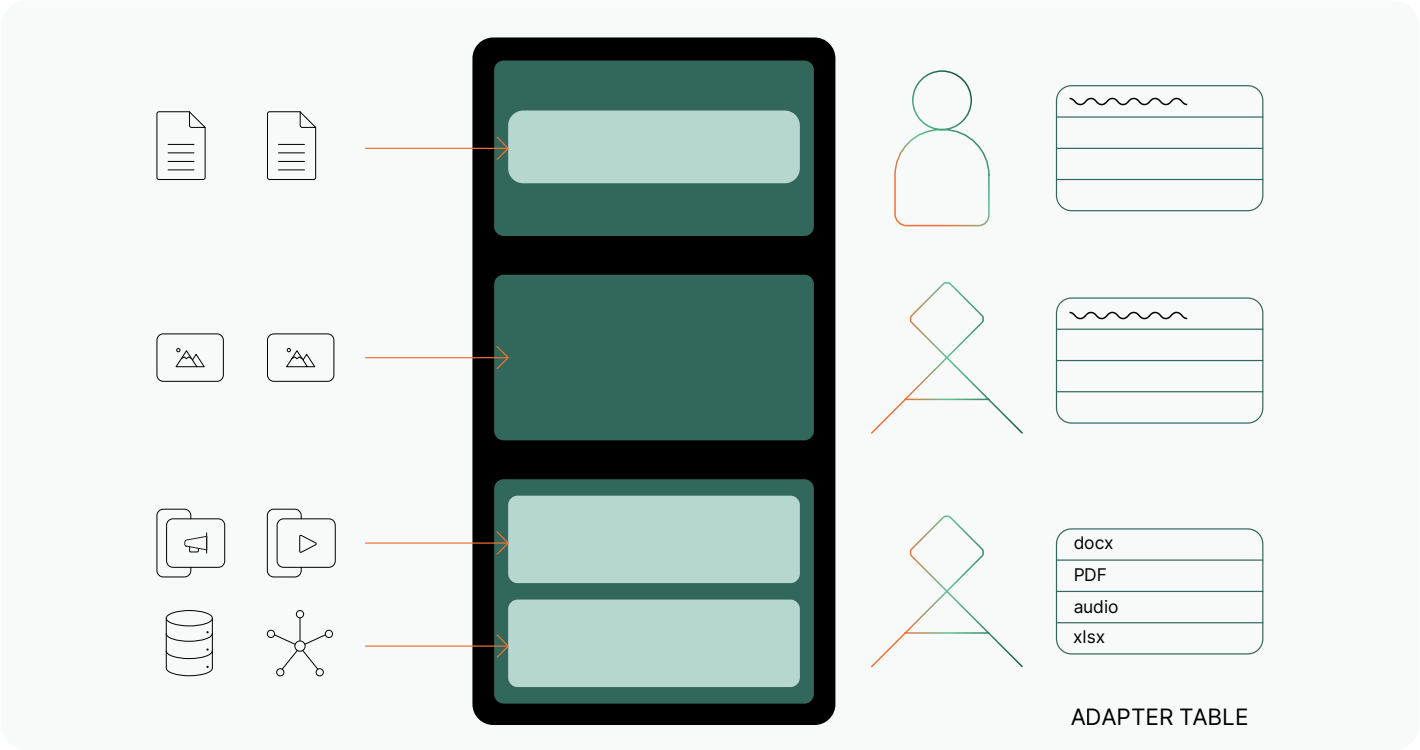


Figure 15. Based on agent type, different adapters are employed to feed information from the global prototype memory and its related persistent storage to different agent views.

Intelligent Operating Systems

Example Prototype Structure

Similarly, for a programmer writing an agentic prototype, the data element servers as a data section, whereby different attributes (e.g., `data-type` attribute) specify how the agentic runtime ought to encode what it reads. Fundamentally, `data-type`, in our case, specifies to the agentic runtime how the textual contents of the data element should be encoded into an object, while the data-adapter specifies how the object should be rendered to the viewer/executor—the agentic runtime mediates between agents (those which write prototypes and those which execute them).

The most fundamental leap that may be difficult to intuit is that the concepts of source control, development and execution are melded into a single system. This way, the agentic runtime becomes the browser, the IDE and the execution runtime, all at the same time—it is an interface for all agents. In this constellation, the user is as much of a program as a program is a user: they are both effectively agents mediated by the agentic runtime, and, if I may extrapolate to the final version, the operating system.

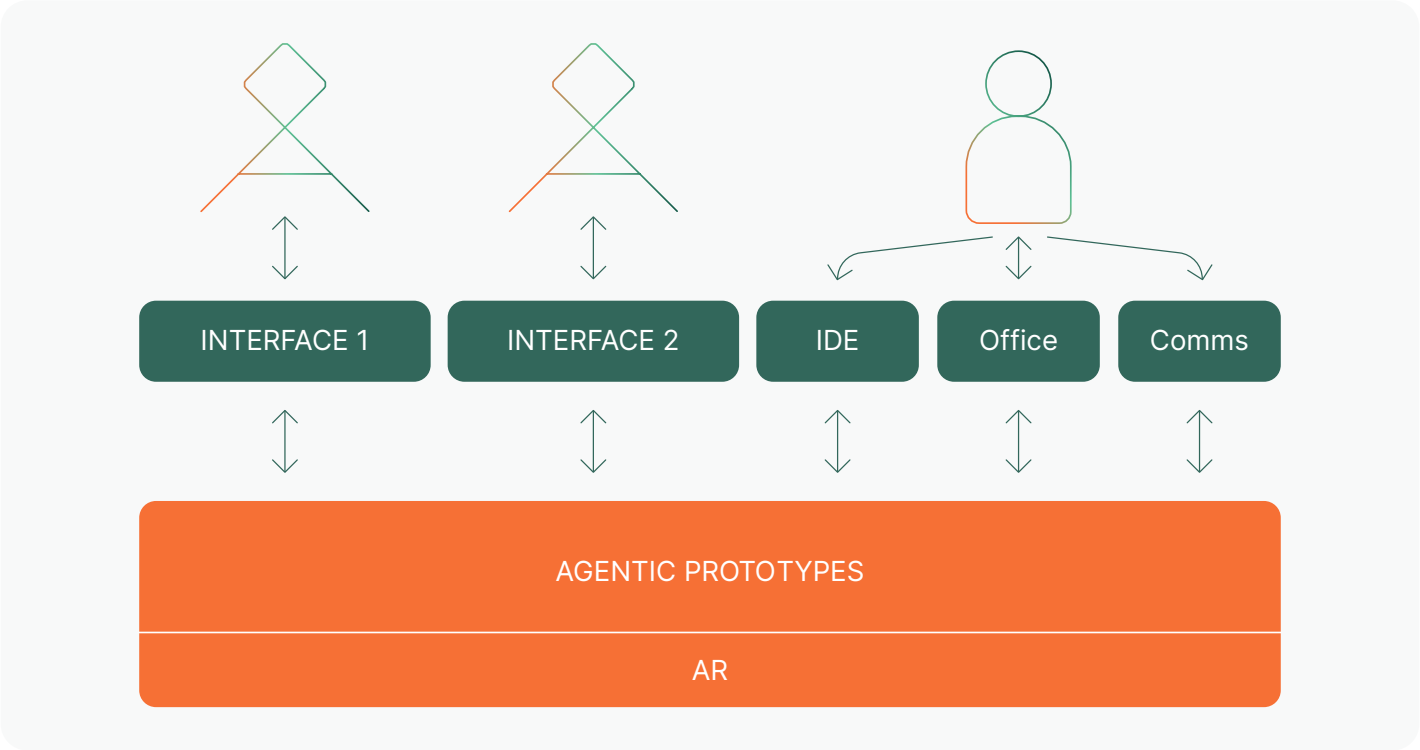


Figure 16. All agents are accessing the Agentic Runtime through adapters—human agents through commonly known software or UI components and digital agents through digital interfaces, APIs and specifically written API components and wrappers.

In this way, the operating system becomes the platform for agent communication and task execution, effectively melding multiple forms of intelligence by virtue of communication moderation—human intelligence becomes infused with machine intelligence, mediated by, at first, machine intelligence. It is worth noting here that, in the idealized future scenario, the entirety of an intelligent operating system, such as this one, is written as an agentic prototype (or via an agentic language, similar to the one we are presenting) and thus the mediation itself may be done by a blend of human and machine intelligence. For now, we will constrain ourselves to the practical example of our own experiment.

Instructions, code and messages

In our toy implementation, we distinguish between deterministic and non-deterministic code execution and, for that reason, between code elements meant for direct execution by the appropriate interpreter and `instruction` elements meant for model prompting. Both serve a similar purpose but instruct the agentic runtime to route the code/prompt to the appropriate interpreter and we treat them as executable elements.

Both element types allow for specifying a kind of method header, or entry point, by which the same element may be called as a method from other executable elements, regardless of type. Most importantly, the fact that an element may be treated as a function allows us to create a message receiving platform, as any executable element may be registered as an interrupt handler, much like it would be the case in a conventional operating system. This way, agents may communicate with one another via message passing moderated by the agentic runtime.

However, we do not specify distinct message elements, and instead opt to treat any element inserted into the running agentic process as a kind of a message. In fact, as we have already discussed, elements are available to executors through views, which typically do not include the entire prototype—the entirety of the memory contents managed by the agentic runtime may be considered a single large agentic prototype of which the developer's starting prototype was merely a view—but they may be decorated with attributes designating their target agent (e.g., `target="receiver-id"`). This way, when an LLM outputs a correctly formatted element with a target attribute, it is immediately routed into the view of the receiving agent and an interrupt procedure (the relevant `code` or `instruction` element) in the receiver's view is invoked. In fact, regardless of whether the element was produced by an LLM, constructed through executing C# code, or typed out by a human developer through a negotiated interface directly into the element body, the agentic runtime will treat the element as a message and route it accordingly.

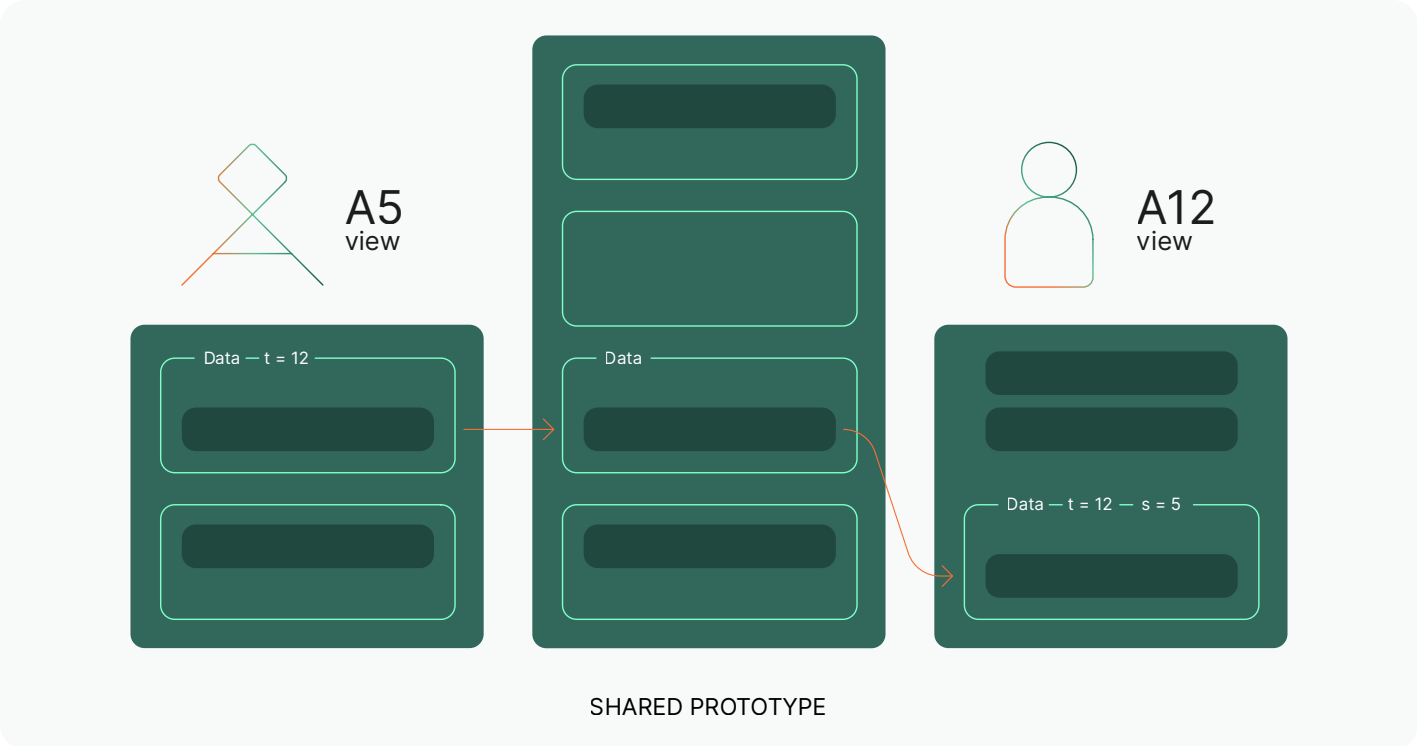


Figure 17. Turn-based conversations are a subset of XML-based distributed agentic computation. Inserting elements with designated target and source identifiers allows the Agentic Runtime to forward messages encapsulated into XML elements to target agents' views.

Furthermore, any element, be it data, instruction or code, may be treated as a message, if appropriately decorated with attributes. In some sense, message passing is nothing more than altering the view of the target agent to include the shared element. The element is inserted into the global DOM available only to the agentic runtime but shared across the agents participating in the message exchange.

To avoid needlessly cluttering the discussion, I will only briefly mention that in the toy model presented here, we distinguish code and instruction elements which ought to be executed immediately (marked with the `execute="true"` attribute) and elements which are registered as methods. We also identify the agentic runtime itself as an agent to all other agents by providing a predefined identifier by which all agents may recognize automated system messages. Furthermore, system messages and logs are automatically added to agents' views when code segments are executed, when errors arise, when messages are received, when context is contracted to retain the window etc. These specifics are outside the scope of this discussion but are nonetheless necessary for the full operation of the toy agentic runtime, and, by extension, the hypothetical intelligent operating system.

Definitions, processing and reasoning

Given that models, as well as humans, perform best when some time is given for either sketching out a solution or reasoning through the steps (e.g., with chain-of-thought reasoning in LLMs), our toy model of an intelligent operating system must account for this kind of processing. Aside from the regular maintenance of the stack and managed memory for deterministic code execution, whose state is revealed to the executor through data elements encapsulated in the view, a second kind of processing occurs in the form of reasoning tokens or hidden reasoning text.

Our toy model allows for such processing by making use of transient `process` elements which, in essence, contain fragments of the thinking process before the final output. For example, an LLM may engage in chain-of-thought reasoning before outputting the final XML element, and this reasoning is, by definition, encapsulated within the `process` element, whose contents the agentic runtime ignores and hides from all views but the executor's. These can be thought of as "internal" to the agent, while the final output is being planned for and worked towards.

We have found that certain functionalities of these toy models cannot be implemented with the current generation models without specific fine-tuning, most importantly fine-tuning for prioritization of two types of elements: `process`, already discussed, and `definition` which serves the function of the typical LLM system message. We have found that encapsulating definition elements between the typical `<|system|>` tags

elevates their importance sufficiently to produce desired behavior in many situations, reducing the need for fine-tuning. However, as argued multiple times before, this constitutes a hack and is, effectively, an alteration of the model's designed behavior. Thus, for a system like this to be truly general, it needs to be trained from the beginning to allow for such functionality, as a turn-based conversational mode is simply insufficiently general.

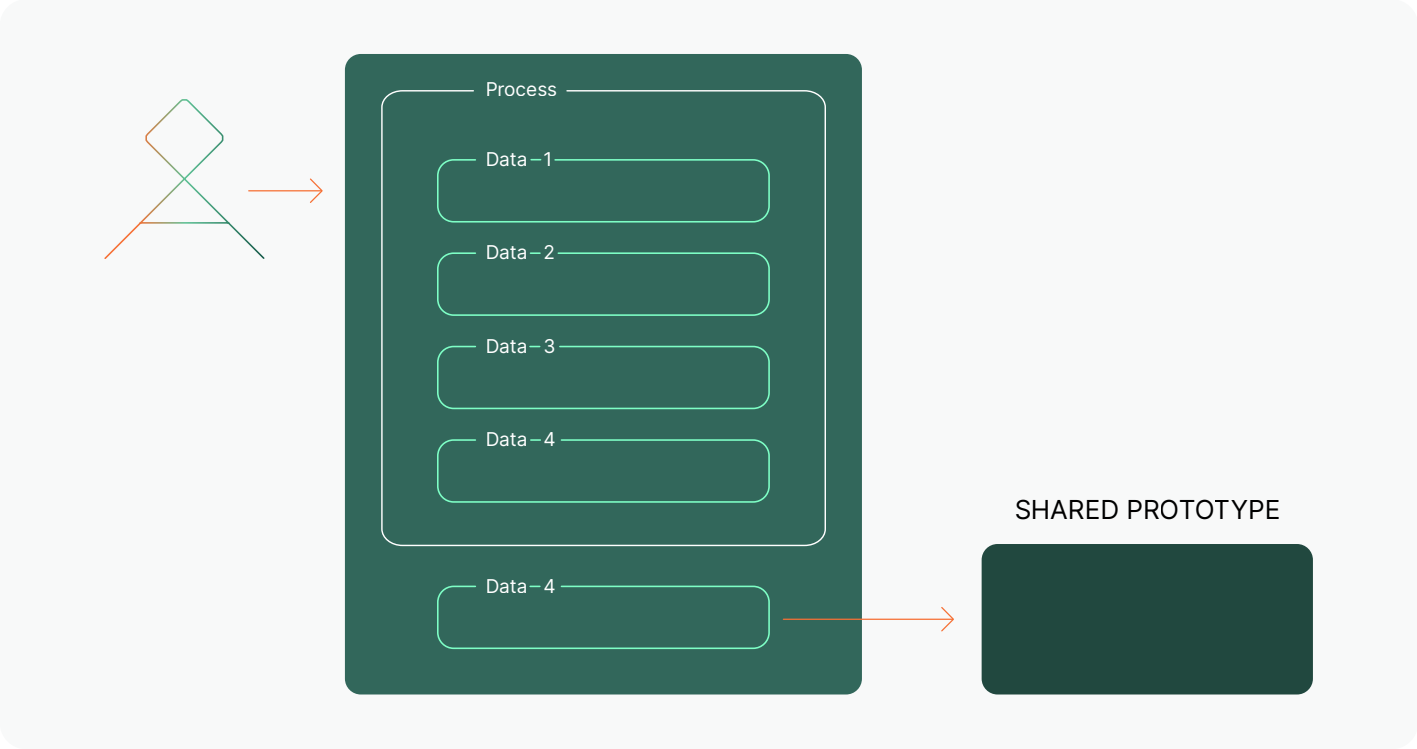


Figure 18. Agents require thinking before committing elements to the global prototype memory. Specific process tasks are used to facilitate thinking and reasoning, without incurring additional load on the Agentic Runtime. Agents only commit the final result, while the thinking artefacts may be marked as transient.

Clearly, we introduce the `definition` elements for the purpose of describing the very contents of this explanation to the models which will operate on the XML data. Furthermore, these `definition` elements may contain specific instructions and modes of behavior that distinguish between different kinds of agents, but they, nonetheless, remain to be akin to system messages in their basic functionality. The basic system description, outlined here through this text, is much more effectively added to the model through fine-tuning, but the need for agent-specific definitions still persists, as they provide behavioral instructions to the executor and may be used to either constrain or expand the set of agent capabilities (e.g., some agents may not exchange messages or even execute code, but execute simple text

Example Prototype Structure

transformation operations).
In effect, a single agentic prototype may entirely consist of a single code element specifying a C# method to be invoked as the agent's body, or it may make use of the full functionality outlined so far. The ultimate aim is to have the entirety of the intelligent operating system be written in the very same agentic language and executed as a single agentic prototype by the operating system's agentic runtime.

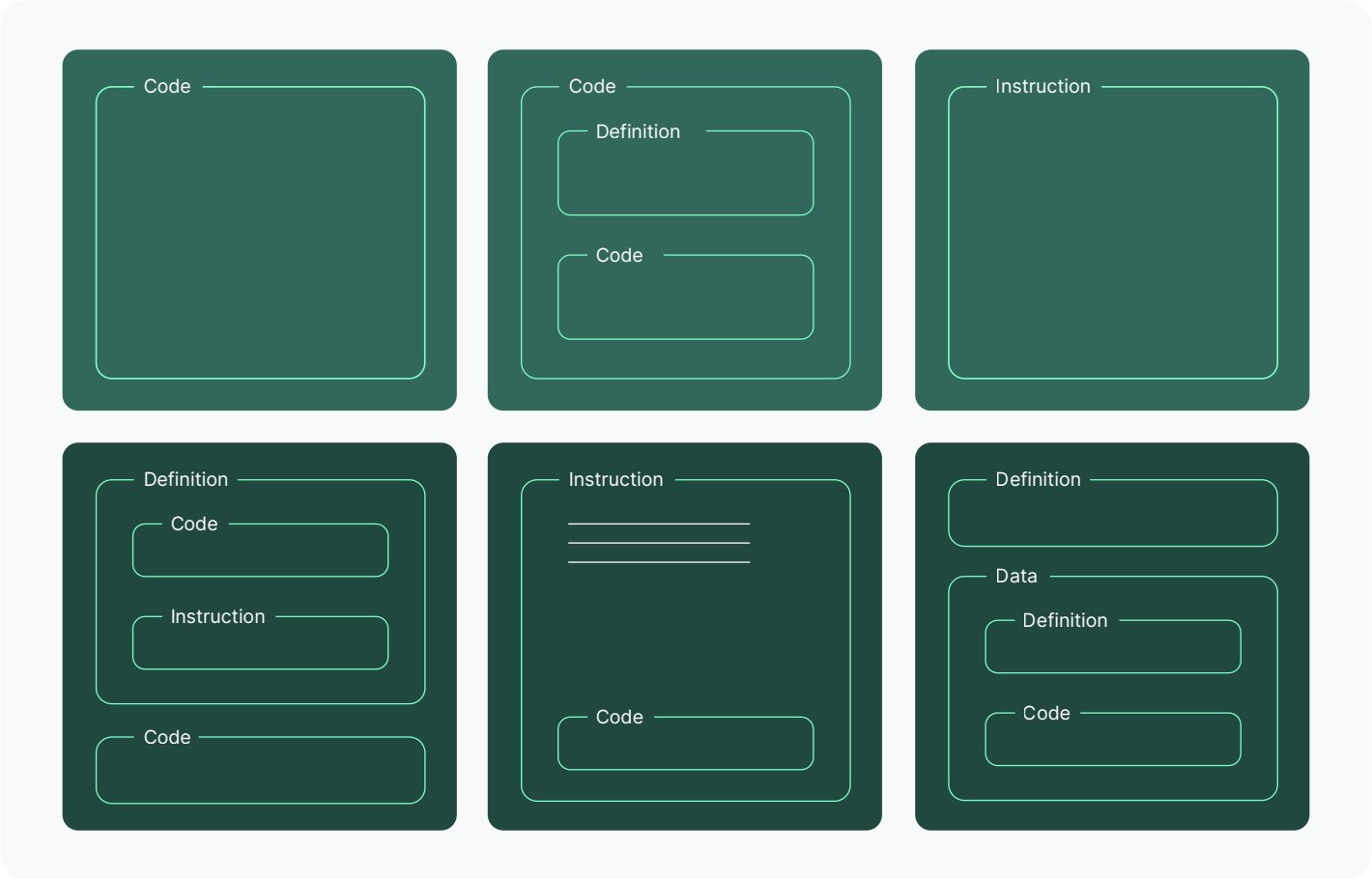


Figure 19. Examples of different agentic program prototypes. Pure deterministic code, pure natural language instructions, pure data, and hybrids of the three all constitute valid agentic program prototypes.

Tasks, schemas and learning

Finally, we arrive at the intelligent operating system's learning capabilities. For any system to be considered intelligent, it must be able to learn and adapt to its environment. In the case of our toy agentic runtime, we must provide a mechanism by which the system can alter its own behavior and adapt so that it uses as little energy as possible while providing higher quality solutions.

In a similar fashion how a human being, a generalist, might undertake a single role in a company and have tasks assigned based on that role, we want to

Intelligent Operating Systems

Example Prototype Structure

design our intelligent operating system to be able to compartmentalize work into roles appropriate to specific classes of tasks. This way, the operating system provides a platform for automation and integration of intelligent agents into a single agentic system. Only by abstracting the human and digital user into a higher-order concept of an "agent" can the entire system become agentic.

Our definition elements already provide a mechanism to mimic human role-based operation, by enabling agent description and specialization. This way, general-purpose LMMs can exhibit specialized behavior, constrained by their definition—the definition which describes what subset of the full functionality they are to undertake as part of their operation—and instruction/code/data elements encapsulated within the agent's view, providing, in the metaphorical sense, tools with which the agent can operate. In a more concrete sense, the way an agent uses any tool is simply by invoking method calls on the agentic runtime's system API. Whether a "tool" is written in C#, JavaScript, is a prompted LLM, or is, in fact, a human receiving a Teams/Slack message, is entirely irrelevant from the caller's point of view—they are merely invoking an external agent through their view and based on their operational definition. For the human operator, the definition elements are simply rendered into their job description and, in the ultimate version, their legal contract.

At a given moment, each agent is ideally allocated a task which corresponds to their role and definition. In fact, an agent may, as part of solving their assigned task, run external agents and assign subtasks to them, based on their definitions and roles. An agent may construct new agents and commit them to the agentic runtime for execution. Effectively, this means that during execution an agent, aside from its definition, must have an assigned task. In our toy system, I used `task` elements to designate the current task contexts. Each `task` element within the current view defines one of the ongoing tasks for the current agent. Note that, since we are using XML to create a kind of runtime DOM for each agent, elements (e.g., data, code and instruction) may be nested within the task element to provide further context and even external tools by exposing agents (and, consequently, methods) which may be invoked through the system API. Thus, a `task` element is a task-specific extension of the `definition` element, which exists within an agent's view until a task is completed by the agent (by issuing the relevant API call).

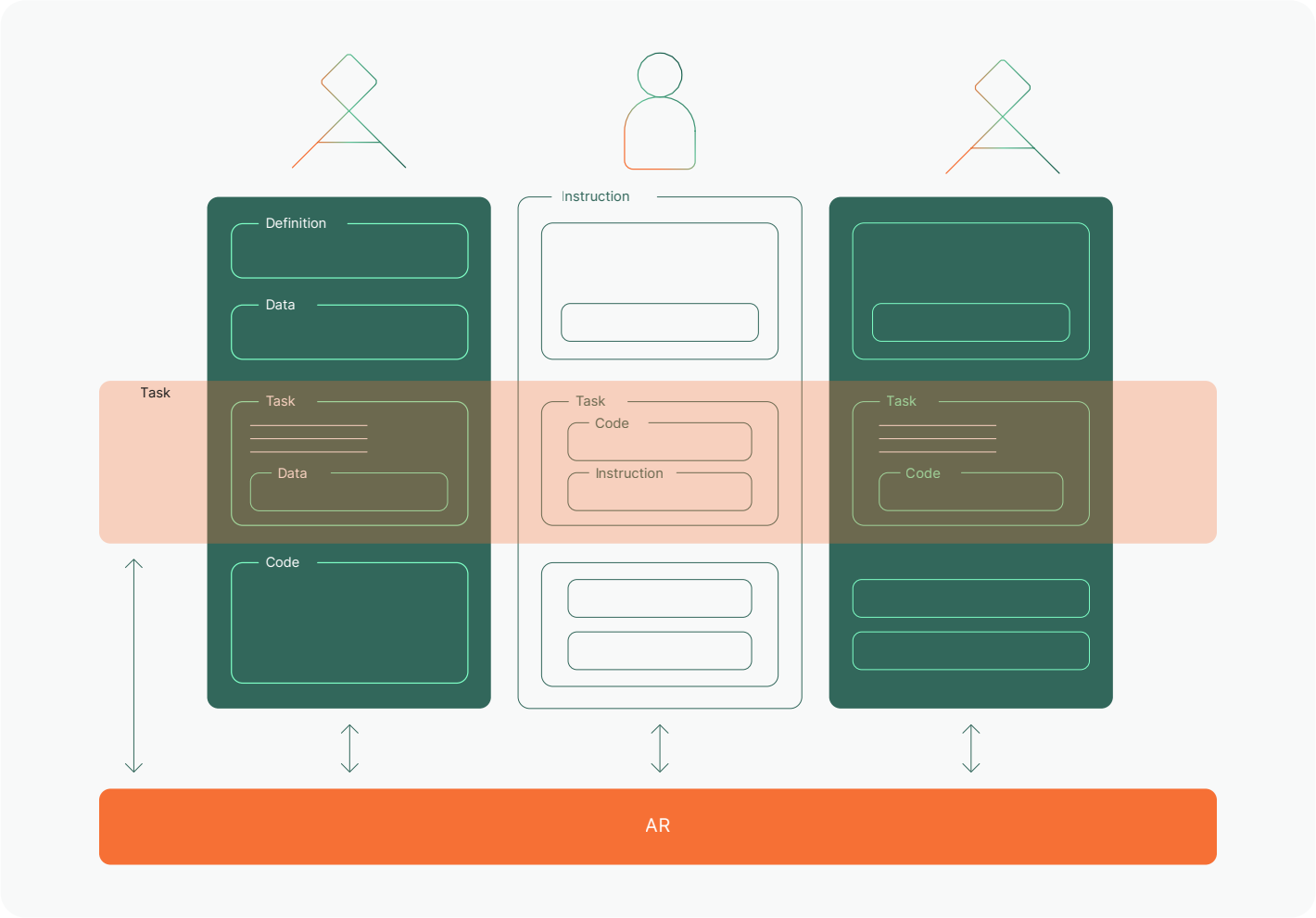


Figure 20. Tasks are temporary definitions managed by the Agentic Runtime, existing across multiple agents until task completion.

Intelligent Operating Systems

Example Prototype Structure

The entirety of our toy system API is not of relevance here, but we should note that a basic system should at the very least include methods for DOM manipulation (e.g., setting content, inserting and removing, and modifying attributes), running agents and invoking their methods, defining and completing tasks, and memory and access management. However, one important system function which is germane is a rewarding function.

Because we want to measure relative success of executed tasks, we want each agent to be able to inform the agentic runtime about the outcome success for each task it runs. If a user runs a task, when the task is completed, be it by another user or by another agent, a kind of rewarding interface is presented through which the task provider can rate the quality of the solution—be that document translation, a coding solution, a management report, or a sequence of cleanup operations on some existing piece of data (anything that might be framed as a task completion artefact)—thereby providing a reward. However, in our conception of an intelligent operating system, we do not reward agents (or users)—we reward tasks.

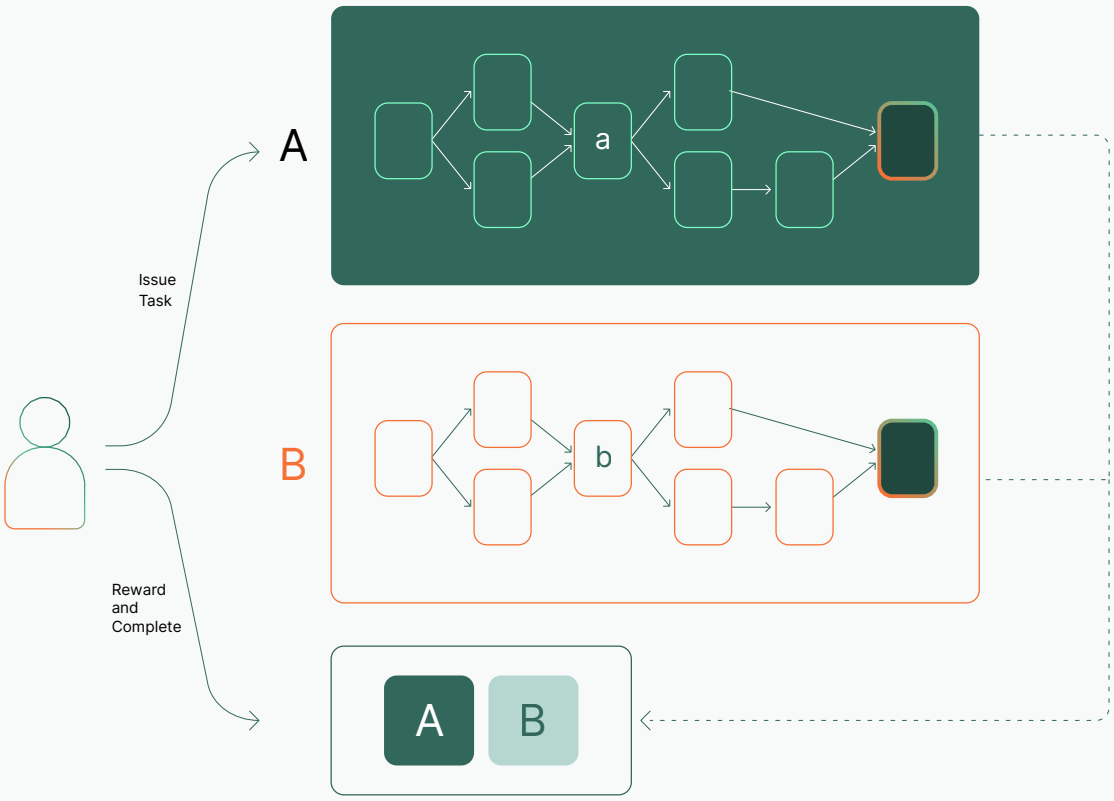


Figure 21. Tasks are completed and rewarded by the issuing agents. Agentic Runtime performs multiple runs of the task, with variations in participating agents, allowing the issuing agent to perform an A/B test and allow system self-improvement.

Intelligent Operating Systems

Example Prototype Structure

In effect, a task is simply a transient context which binds multiple agents together in a sequence of operations with an expected outcome. Only the agent with ownership over the task can mark the task completed (and set completion status), but agents are allowed to transfer ownership, as well as spawn subtasks and issue them to other agents. This way, a task issued by an agent creates a temporary state machine comprising of multiple interacting agents, until the machine terminates, and the shared task context is what remains as the task completion artefact.

This approach creates the basis for schematic learning and self-revision. Simply put, those agents who more often partake in tasks rewarded highly are more likely to be selected for execution as part of future tasks.

Schemas and Reinforcement Learning

The fact that an agentic program is simply a piece of XML-formatted text and that a running process, or view, is formatted in exactly the same way, blurs the distinction between the traditional program and process. In fact, any snapshot of a running agent may be taken as an agentic prototype. In essence, any view containing a definition and at least one executable element may be treated as an agent.

By design, this allows for alterations and variations on the same agent prototype. In the same way one might use A/B testing to see which single prompt or prompt sequence may be performing better against a set of metrics (similar to how this is done in, for example, Azure Copilot Studio), we allow variations on agents to be included within a task context. Essentially, every modification requested by agents is tracked by the agentic runtime, similarly to how a source control platform might track file changes. The key difference is that the changes are usually discrete and only effected at XML element level, so the runtime can easily track prototype versions, as well as agents (including users) who made the alterations. For the purpose of implicit A/B testing, the agentic runtime maintains multiple versions of prototypes simultaneously and tracks their relative rewards. This way, when a task is assigned to an agent, the agentic runtime will execute multiple agent state machines simultaneously and provide the caller with one or more solutions. When receiving task completion artefacts, the task issuer must either accept one of the proposed solutions or request a new attempt (in cases where the presented solution is unsatisfactory or a failure). Whether the task issuer is a human user, or a digital agent is irrelevant from the agentic runtime's point of view—the feedback provided by the issuer is an essential part of the task execution life cycle and it is used by the agentic runtime to promote prototypes or discard failing alternatives. In effect, a form of reinforcement learning is in effect, selecting for the best performing prototypes.

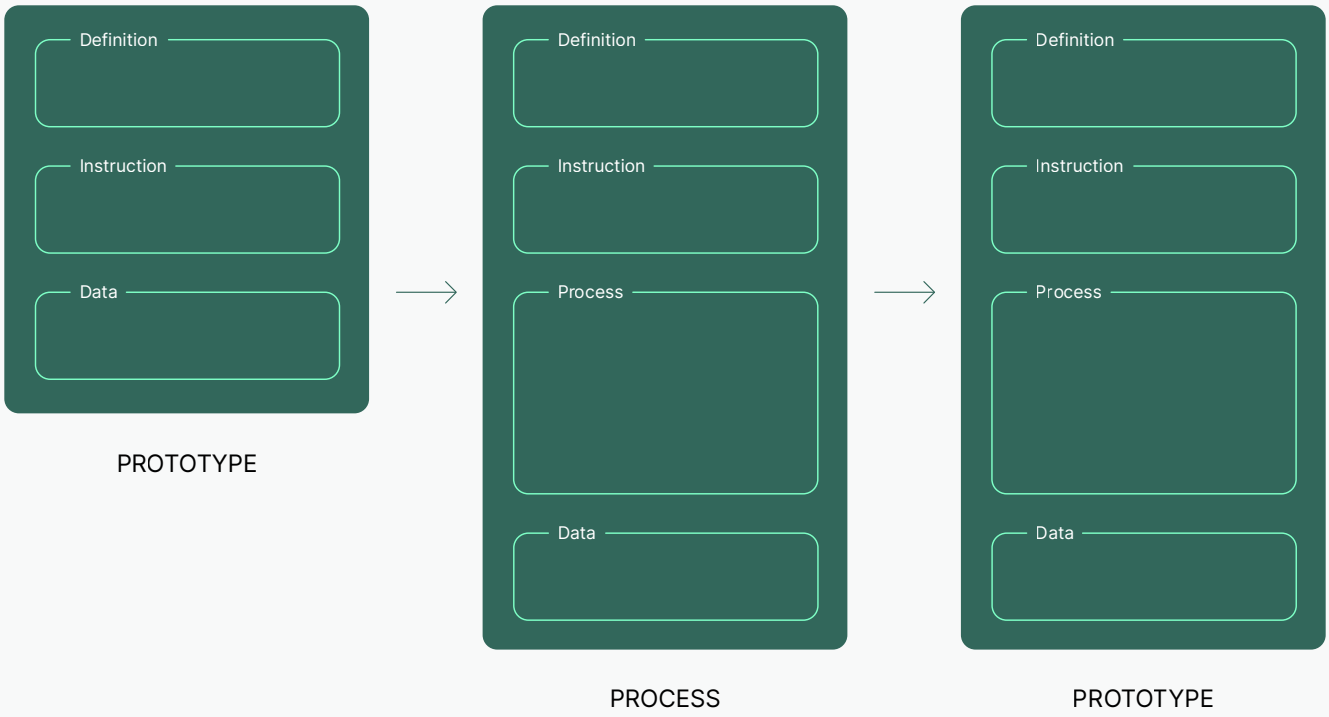


Figure 22. Programs and processes are less distinct than in traditional operating systems. An agent might turn a process into a prototype, edit its contents, and run it as a new process.

Intelligent Operating Systems

Schemas and Reinforcement Learning

Over a sufficient number of runs, certain agents (prototypes) will be selected for diverse sets of tasks, others will be confined to specific tasks, while worst performing ones will be entirely removed from the global prototype pool. Over time, the successful execution chains will be preferred and may be used for further model fine-tuning. The more examples of successful chains of execution are provided to the model during fine-tuning, the more likely those chains are to repeat in the future. In fact, this is how the o1/o3 and family of models was trained for reasoning. I am merely proposing a more generalized approach.

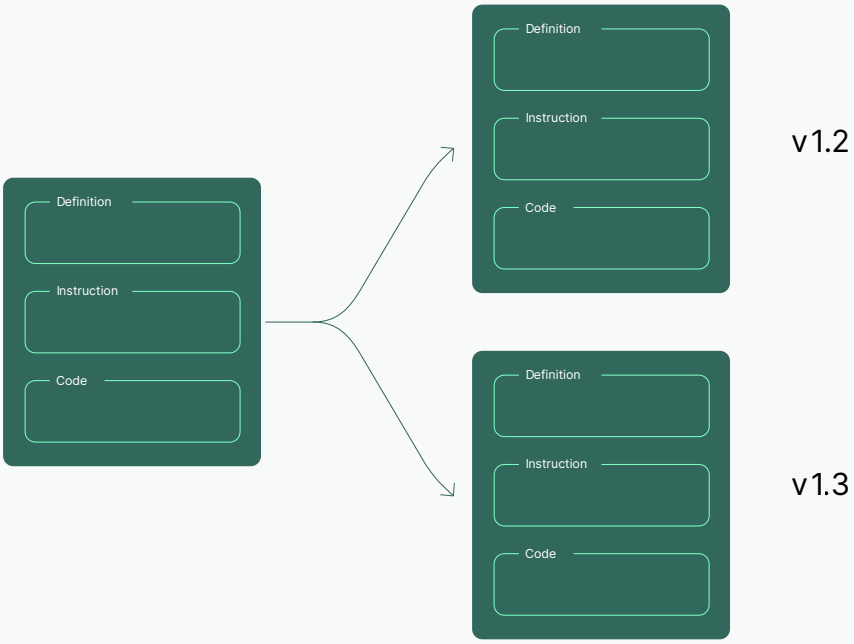


Figure 23. Individual programs (agentic program prototypes) can be branched and created variations on. These variations may be used for A/B testing and code improvement.

By this point, an attentive reader will have noticed a clear relationship between how our toy intelligent operating system operates and how an effective company operates. Our aim here is to illustrate that the future is almost inevitably a convergence of the two modes of operation.

Full and partial automation

One might pose the question of the agentic runtime's reliability with regard to evaluating nuances in task performance, as well as the rewarding algorithm itself. However, in the ultimate implementation, this algorithm itself would be subject to revision and rewarding.

However, when a human user is treated as an agent within the system, the user might be exactly the agent to be invoked to make the subtle comparison, until the algorithm arrives at a more suitable digital agent. The A/B testing previously discussed does not need to be between two digital agents or between two human agents, but instead a way to compare their relative performance. In fact, multiple comparisons against the same successful human agent constitute a kind of transfer learning from the human mental model to the digital schema.

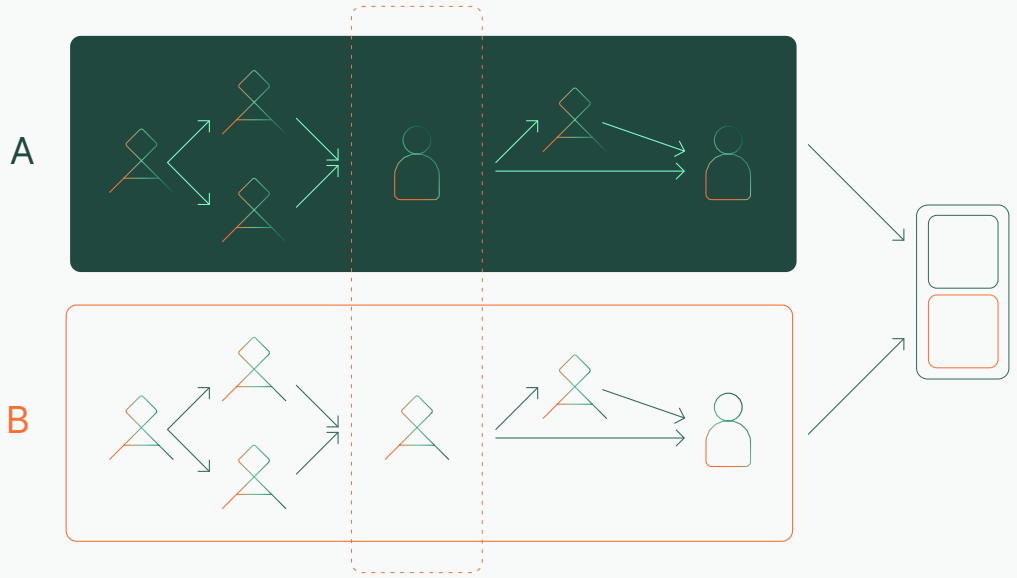


Figure 24. The Agentic Runtime does not distinguish between agent types. An A/B test might only differ in a single agent—human in one variation and digital agent in another. The better performing chain is more likely to be selected in the next iteration, as is the case with human-human variations and tool-tool variations in companies today.

Intelligent Operating Systems

Schemas and Reinforcement Learning

The question of so-called full automation here is resolved by the very existence of the agentic platform which encapsulates both human and digital agents and mediates their communication. Tasks in which human agents are clearly better are delegated to human agents and vice versa. Thus, as time progresses and more digital agents are derived by the agentic runtime, both through actions of digital and human agents, the more work will be handled by the digital kind of intelligence.

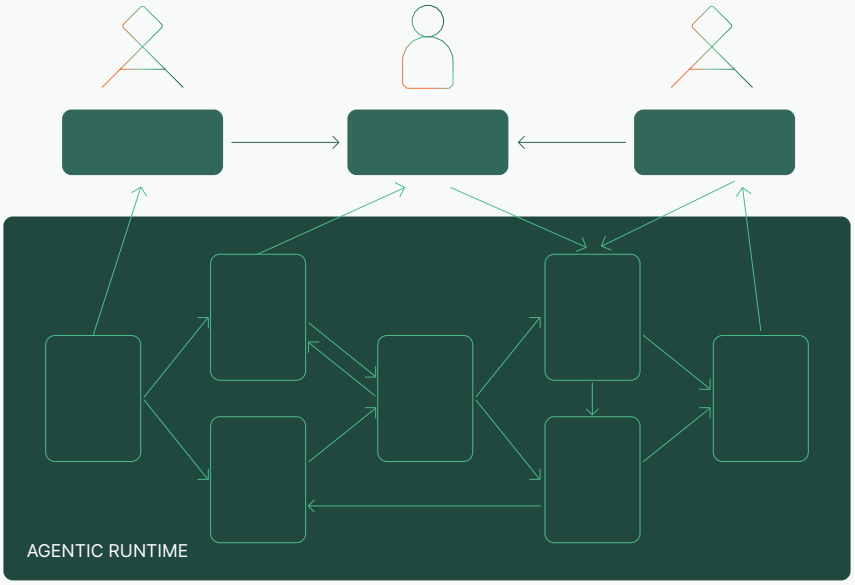


Figure 25. Eventually, the entire system is written in agentic program prototypes, rather than a specific programming language. It is written in the very hybrid language it is made to run.

Intelligent Operating Systems

Schemas and Reinforcement Learning

Future Vision for Intelligent Operating Systems

The ownership component of the agentic runtime—the intelligent operating system—is what allows a human user to still maintain control of the system, even in cases where a digital agent might be superior in performance.

If it remains to be the case that humans are always the initiators of tasks (i.e., the source of purpose for the organization—the integrated intelligent operating system—remain to be humans), then the digital agents cannot, by design, take control.

Whether it is likely that a highly intelligent cluster of agents would, through various means of manipulation, attempt to affect the human owners to relinquish control in favor of convenience, is outside the scope of this discussion. If the same tendency towards energy minimization which predisposes humans to laziness is added to the rewarding system, a stable balance in which a similar number of human and digital issuers and executors is maintained may be feasible, but this is well outside the scope currently testable by experimentation on toy models, since current generation models do not possess nearly the necessary subversive capability.

Future Vision for Intelligent Operating Systems

Although our example was overtly technical, the comparison between an operating system and a company would have been extraordinarily difficult, without resorting to crude metaphor. Our toy model of an intelligent operating system clearly indicates that a more general approach to agentic systems is possible, while maintaining the conversational aspect that the models are currently geared towards. As argued before, implementing models to make use of XML structuring instead of message-based division provides a more general usage model and the current generation models show clear capability of using the same XML-based scheme for turn-based conversation.

Of course, we have observed multiple failure cases of general intelligence, most notably when nesting agent definitions inside `data` or `process` elements. The level of necessary theory of mind to distinguish between nested definitions and instructions of other agents and agent's own operational instructions is still beyond the current generation models. Although they do succeed in many of the tasks, their performance is unpredictable and not yet reliable enough to be implemented within a

Intelligent Operating Systems

Future Vision for Intelligent Operating Systems

rewarding system or as their own agent-constructing systems without human supervision. However, our experiments clearly indicate that this capability is on the horizon, if appropriate training methods are used.

Most importantly, however, we can see a clear pattern of role abstraction: a fully automated company is, in effect, indistinguishable from an intelligent operating system. Whether it is implemented through the exact framework outlined here or by means of some commensurate approach, a fully integrated intelligent operating system will abstract the user's role as an agent.

A future in which agents are negotiating their communication interfaces, facilitated by a singular operating system, dispenses entirely with the traditional notion of the frontend and removes the barrier between the operating system and the user-mode application. Furthermore, the boundary between digital software and human cognitive software—minds, to use a cruder metaphor—is blurred to the point that the distinction between the issuer—the agent/person who invents the task to be solved—and the executor—the agent/person who performs the work is all but gone.

The discussion about intelligent operating systems and automation cannot be separated from the discussion about human autonomy, freedom, and group coalition from which companies are produced. In effect, the more integrated the company, the more singular the forms of intelligence comprising it. We are inevitably becoming part of the software we are building. The future in which society is automated is the future in which legally binding contracts are increasingly expressed through code. Although that future may be relatively far, for mostly infrastructural reasons, its inception has clearly already begun.

Representation and Meaning

Whether we analyzing how human-to-human communication uses language as the means of distributed social computation or we are investigating how information is distributed amongst different digital agents executed by an intelligent operating system, or we are assessing how multiple neurons in an LLM encode the same concepts through superposition, we are, in effect, discerning how single concepts are split and distributed across a system's components.

In order to understand how a piece of information may be distributed, we must understand in which ways it may be split and merged. The issue then becomes that of which pieces of information may be considered *atomic* and how *compound* information can be expressed through atomic components. In effect, an atomic piece of information cannot be split into constituents and, in a distributed system, may only be stored on a single node, while compound information, in principle, may be distributed across nodes.

However, given our preceding discussion regarding language and distributed cognition, the problem is greatly complexified when we attempt to constrain the notion of information to "what may be expressed in language", which we must do in concrete applications leveraging language models and textual storage in general (which is the dominant way in which information is stored in any organization managed by humans). Ultimately, the discussion will converge on one of the outstanding problems in mathematics and linguistics in general, namely that of how the real-world fact and logico-mathematical symbolic apparatus are related. Such inquiry borders on philosophical (Wittgenstein, Ogden and Russell 1981) and the space afforded by this treatise is not conducive to such discussion, so we will constrain ourselves to more practical matters.

What is relevant for the practice of working with LLM- and LMM-based systems, such as our intelligent operating system framework or company internal organization in general, is how text (and other modalities relevant to LMMs) may be used to efficiently represent information so that it is available to all participating agents, including users and digital systems, such as chatbots or agentic pipelines.

Distributed Information Retrieval

We have seen demonstrations of “needle” recall (i.e., needle in a haystack search) with multiple different LLMs (Google Gemini Team 2023). The retrieval in question assumes atomic pieces of information being retrieved. In other words, the model in question is tasked with retrieving a specific piece of information hidden somewhere within the “haystack” of text (as well as video and audio). However impressive the performance of the Gemini 1.5 model family is on this task, the needle-in-a-haystack retrieval problem is somewhat simplistic in the sense that the queried piece of information is atomic.

For retrieval to be successful in this scenario, the retrieving model does not need to reason about the piece of information itself beyond its difference from the rest of the content contained in the context window. The real question of model reasoning capability arises when a compound piece of information is split into its constituents and distributed across the context window into multiple needle fragments. In order to retrieve this piece of information, the model must have a better representation of what is being asked for and how parts of the content within the context window relate to one another. To synthesize back the distributed needle, the model must identify fragments across its context window as well as be able to connect them back into the needle.

A clear problem arises here, one quite salient to the question of linguistic concept representation: not all piece of information present the same difficulty for retrieval. If the piece of information is more embedded in the context (i.e., more different in structure, formatting and general linguistic pattern from the rest of the content), it is easier to “spot” by the retriever then would be the case with information that is a natural part of the content. Furthermore, retrieving specific strings requires less intelligence (i.e., less generality and less informational coupling), then retrieving a piece of information that is less directly and more metaphorically related to the query. In a sense a simple block-by-block string search algorithm utilizing embedding similarity comparison is much faster and equally accurate in this scenario. However, the more embedded the piece of information is in the context, the more difficult it is to retrieve, as its retrieval requires a kind of reasoning about what is being retrieved. For example, retrieving a split MD5 hash string from a corpus of poetic works is much easier than retrieving the same split string from a collection of MD5 strings. Additionally, it is much easier to retrieve a split MD5 hash string than a split verse from a poem, regardless of the context. Part of the reason for this difficulty is in the depth of understanding necessary to meaningfully split (and merge back) a piece of literature in comparison to simply recognizing the pattern of something like an MD5 string.

Representation and Meaning

Distributed Information Retrieval

In other words, the distributed piece of information being retrieved is characterized firstly by the degree to which it is atomic (i.e., how much it may logically be split into components and how reversible the split is) and secondly by the degree to which it is integrated in the context. The more integrated a large piece of text is, the more its parts are inseparable from one another. In that regard, when the needle piece of information is embedded with the text, the needle itself becomes an inseparable part of the containing text. The more integrated the material, the more the text itself becomes an atomic piece of information.

In other words, the retriever of the needle is attempting to find the "seams" which bind a needle with the containing context, but the more integrated the needle, the more invisible the seam and the more atomic the container. Furthermore, the more compressed the textual representation (i.e., the more informationally packed, the less redundant the text), the more difficult the task of retrieval of its parts becomes. An attentive reader will notice the relationship between measuring representation complexity and the intelligence needed to compress the information to such a complexity level. As stated earlier, compressed information is indistinguishable from noise, and, hence, retrieval of any part of it requires a decompression procedure—reasoning. Reasoning is what enables compounding of information into more integrated units, as well as what enables unpacking of specific information from an integrated unit. The product of action of intelligence is a compressed representation.

In order to measure an agent's ability to retrieve information, we must, at the same time, assess its ability to reason about the information being retrieved. Retrieval via string matching requires neither understanding nor intelligence, while retrieval via reasoning against the context requires both. The issue at the level we are concerned with (namely that of retrieving and inserting information in human-sourced text) becomes that of quantifying the amount of reasoning necessary to retrieve or store a piece of information. In other words, how do we quantify the difficulty of a retrieval task?

In effect, we are coming back to the question of intelligence: the difficulty of a retrieval task is determined by the minimum required level of intelligence to extract the desired piece of information. In other words, the generality of the information encoding algorithm must be matched by the generality of the decoding algorithm. Essentially, if the decompressor knows exactly how to retrieve a piece of information from a string (e.g., there is a specific extraction or search algorithm for retrieval), no computation is necessary to discover the algorithm itself. However, if the decompression procedure is not known, computation is necessarily allocated in order to discover the algorithm itself. This generalized computation we typically refer to as reasoning, which results from the action of an intelligent agent.

In other words, an agent which is more successful at retrieving needles for whose retrieval it was not trained is more intelligent (i.e., general) than an agent already trained for retrieval. In some sense, a deterministic decompression algorithm which reliably retrieves information from texts structured exactly according to a compression/decompression standard

defined for the specific scenario in which the algorithm is used is a system which has been overfitted during training for that specific scenario—a specific decompression algorithm is an overfitted general algorithm. On the other hand, the most general algorithm (including its state) would be indistinguishable from random noise and likely require substantially more compute resources, as each decompression attempt would necessitate a complete invention of the decompression procedure, given that such an agent would have no specialization for the case whatsoever (i.e., it would be entirely general in its approach). A fully general algorithm would not distinguish between problems it is applied to.

The issue of generality arises even when constructing our retrieval metrics, since when creating distributed needles, we must account for a needle's complexity with respect to the model, needle's atomicity (i.e., the degree to which a generated needle may be reliably split into fragments), needle fragments' semantic relationships (i.e., fragments must be combinable back into the original needle), the latter two of which may be numerically quantified, while the first is currently beyond direct measurement.

Distributed needle-in-a-haystack metric and metric generality

One approach to evaluating distributed needles is to generate a diverse corpus of candidates and make use of existing models (e.g., OpenAI o1) to split the needle into multiple fragments.

For our experiments, we created a distributed needle generator which attempts to create a set of fragments which may be reliably reconstructed into the original. We attempted to maximize the semantic distance between the fragments and the original and the fragments from one another, while minimizing the distance between the reconstruction and the original.

For each generated needle, we reran reconstruction and checked the average reconstructed needle semantic distance with the source needle. To qualify as a viable test sample, a generated needle must satisfy three conditions: first, it must be reconstructible by multiple models in as many tests as possible, second, it must not be reconstructible from any proper subset of fragments in as few tests as possible, third, the average distance of the reconstruction must be as close to the source as possible.

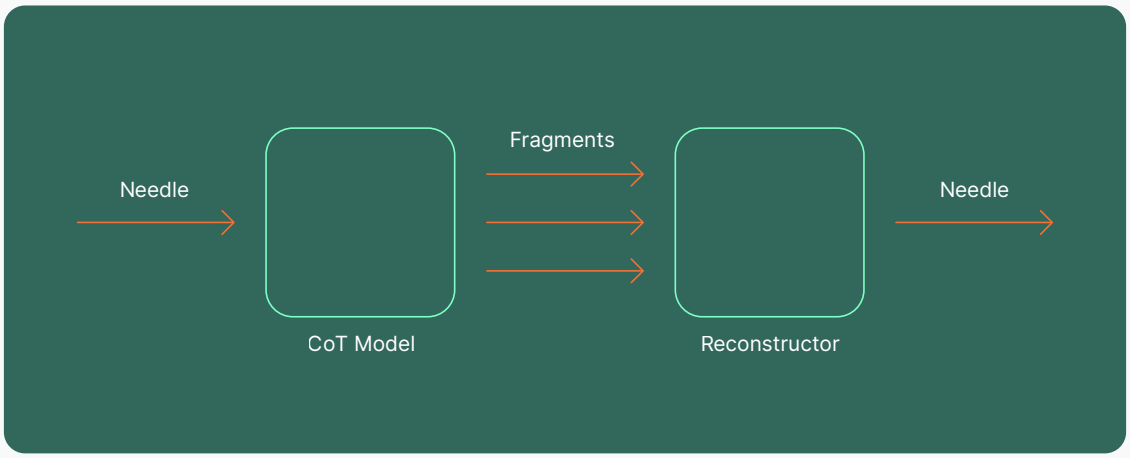


Figure 26. Chain-of-thought models creates needle fragments, reconstructed subsequently into the original needle.
The system must guarantee that reconstruction is never possible from any individual fragment, but always possible from all.

Representation and Meaning

Distributed Information Retrieval

Clearly, for each of the three criteria we must designate a statistical threshold which stands for “as many as possible” and “as few as possible”. To qualify as a valid needle, a generated pair of a needle and its fragments must meet all the predefined thresholds. In our tests, we set these thresholds of statistical significance in accordance with our research budget, but, in general, the thresholds may be tuned to match a desired level of statistical significance.

Nevertheless, the metric as presented is still contingent on the objectivity of our measure of similarity, which, if we make use of cosine similarity comparisons against embeddings of the phrases, are further dependent on the embedding models and algorithms. In essence, the question of language vagueness cannot be circumvented and, in the final analysis, reduces to the foundational problems in logic and mathematics, which are somewhat out of scope for this treatise.

I present this specific retrieval test as an illustration of the main problem we are dealing with when it comes to evaluating intelligent systems, namely the vagueness that comes from the distribution of our representations. The reason we cannot reliably measure or define the performance of intelligent systems is because the very representations used for measurement are distributed between us—the measuring agents—and the system being measured. The more similar our approach to the agent’s, the less we are measuring its intelligence and more its performance on a specific task. The more we recognize the algorithm used, the more we measure the performance of the algorithm. The more we identify the representation in an agent, the more we understand it, the less it becomes general with respect to us.

Representation and Meaning

Distributed Information Retrieval

Integrating Social and Digital Ecosystems

Effectively, our act of understanding a system makes the system less general with respect to us and us more general with respect to it. In this sense, recognizing that understanding a system entails reducing its relative intelligence enables us to see the futility of attempting to measure "general intelligence". We can conclude that, if we are to stay within the domain of practical, we ought to measure performance across more specific tasks—ones for which we can construct metrics which quantify performance in real-world applications—rather than generality of performance. In this sense, the fact of whether the model generalizes or not becomes less important than enumerating which real-world tasks a model is applicable to.

A measure of a model's generality (i.e., its intelligence) will spontaneously emerge as our performance benchmarks generalize. In other words, the more diverse benchmarks we use for evaluating the models, the more the median result on those tests will become a measure of models' intelligence. In some sense, by generalizing and diversifying our benchmarks and the set of problem spaces the models are evaluated against, we will have created an indirect generalized test of intelligence—by pitting our society's generality against the generality of our artificial systems, we will obtain a relative measure of their intelligence with respect to ours.

Integrating Social and Digital Ecosystems

As long as we are aiming to retain a clear boundary between the biological and the technological, true abstraction of social and digital systems will remain inaccessible to us. If we wish to store information in a way which is conducive to reading and manipulation by human and digital agents alike, then those two types of agents must become alike. Otherwise, some form of compromise will need to take place, either in the form of information redundancy (i.e., lower combined intelligence) or in the form agent-specific adaptation or information transformation.

In our example of an intelligent operating system, I advocated for the latter approach in which the data itself is stored in an agent agnostic state, but each agent accesses it through a set of adapters unique to it. For example, in our toy implementation when an audio file encoded as samples was accessed by a user (i.e., a human agent), it was rendered through their machine's speakers, when it was accessed by a non-multimodal LLM, it was rendered into text by an adapter which itself contained a chain of specialized models for audio recognition. In other words, each type of agent is presented with a different rendering of the underlying information.

Representation and Meaning

Integrating Social and Digital Ecosystems

Ideally, stored information would be optimally compressed and agnostic towards the agent type, but in practice this is not feasible, given that all information added to the system is always added by an agent, be it a user or a digital agent. Our solution to this was providing converters from one format to the other, instead of having an intermediate agnostic format.

Furthermore, as a system consisting of multiple agent types scales it will necessitate multiple indexing schemes, each appropriate to the agent type. As long as the distinction between agents is maintained, the system will inherently contain redundancies. Thus, in order to improve a system's generality, all its components must improve their generality. As observed before, the higher a system's intelligence the more scale invariant it becomes. If our aim is to improve an organization's intelligence, then its parts must become more general and more integrated. Whether we are talking about companies as organizations, digital "organizations" consisting of interacting agents, or organizational hybrids, as laid out through our intelligent operating system example, its generality is contingent on the generality and interconnectedness of its components.

Whether information is stored in a centralized manner or distributed across agents is less relevant than whether the information is accessible to all agents constituting the system. In fact, the higher the intelligence of the system, the more connected it is, and the more accessible each of its parts becomes from all others, leading us towards the conclusion that a kind of representation distribution is an inevitable consequence of component generalization. Moreover, as said before, the more general and connected the components, the less componentized they become and, consequently, the more arbitrary the choice of distinguishing between the components becomes.

In current day's practice, however, the generality of companies is far from the theoretical one outlined above. Though society itself may exhibit a degree of generality, its components certainly do not—society remains modular, much like its modules, all the way down to individual people or individual digital models, where we first see the level of coupling conducive to the analysis above.

The future in which companies are becoming more general and integrated with one another is highly untenable, not so much due to obvious infrastructural and political constraints, but for the pure reason that the notion of a "company" as we see it today would almost certainly evolve as society approaches the level of generality in question.

Today, the concept of a company contains a tacit assumption about its modular nature with respect to society—a company provides goods or services of a specific kind and hence constitutes a social "module" of a kind which serves a specific, not general, purpose. A system which is so well organized so as to be general resides on a higher level of abstraction than a company—it is either a society or something coming from an entirely new paradigm.

Representation and Meaning

Integrating Social and Digital Ecosystems

Nevertheless, the introduction and integration of highly intelligent (i.e., general) digital systems—systems that grow, learn and improve in their generality—into corporations and societies will inevitably lead to either differentiation between humans and the system or our assimilation into the system. The former is of less concern here, as it would entail no change in the way we think about process and information organization. The latter, however, poses an intriguing challenge: if we expect to integrate our intelligence with the evolving digital one, our way of life will need to adapt and accommodate to the change, something we have historically always done with the emergence of new technologies (Diamond 2017).

Thus, any company setting its eye on survival ought to prepare for the inevitable convergence of the two types of intelligence. In some sense, wishing to retain the present legal and economic form of our organizations is fundamentally contrary to the integrative future, since companies are made to accommodate to our ways of communication, our ways of storing and interpreting information, and our ways of working, which are biologically and socially determined. In other words, we would be in the way of organizational evolution.

Our intelligence limits the intelligence of the system we ourselves constitute. If we impose our way of information storage and our way of communication onto the system's internal operation, we have effectively made it think like us, thereby limiting its potential to generalize beyond how we think.

In this sense, in order to effectively integrate with the parts of the system which possess a higher degree of generality, without interfering with their operation, we must understand their representations. The more we understand a system, the less intelligent it becomes with respect to us. Our effective understanding of a phenomenon—one which may be used to predict the phenomenon—is a direct indication of a forming shared representation. The more reliably two components represent one another, the more singular they become.

Consequently, to build systems—digital software or social organizations or hybrids of the two—whose intelligence scales well, we must aim to improve the ways in which we store and interpret information, rather than design them for our natural convenience. As companies complexify and their intelligence/generality increases, so will the intellectual difficulty of the tasks assigned to humans who are part of the companies' systems. Those humans unable to intellectually meet the requirements of the system will not be part of it. Increasing the intelligence of an organization decreases its tolerance to lower intelligence, whether the increase is done through integrating high machine or human intelligence.

In other words, an agent can either increase its intelligence and take part in the system, decrease the system's intelligence and take part in it, or relinquish taking part in it. Note that the term *intelligence* I use in this context is somewhat, although not largely, different from the traditional notion. By *intelligence*, I really mean *generality* and *connectedness*. In the context of participating in an intelligent organization, the effective

measure of increasing one's own intelligence relative to the system is how much the individual understand the entirety of the system it is taking part in. The notion of intelligence laid out here is by definition relative, so, by increasing their own knowledge of the system, an agent is effectively becoming superordinate to the system. In some sense, effective applicable knowledge of the system is indistinguishable from intelligence in this case. However, the more general the system being modeled is, the more general the knowledge of the system becomes. Thus, the generality of knowledge (i.e., representation) of the system is, in a practical sense, a measure of intelligence with respect to the system.

For a company offering a specific service within a society, there are various degrees of skill generality present among the agents comprising the company. If such a company is supplanted by agents with higher intelligence, and hence generality, the demand for more generality within the domain will increase, not decrease.

Practically speaking, this has been the case for centuries: modern workers are required to possess multiple skills, unlike the craftsmen of the past, who could afford higher specialization—as society evolved, so did the demand for generality. In modern companies, basic literacy, English proficiency, software and computer literacy, driving skill, legal literacy are all necessary, if not required, skills for any worker, regardless of their specific niche. In effect, workers are becoming more general as time progresses. This is mirrored by a famous phenomenon, dubbed the Flynn effect (Flynn 2009), whereby the average intelligence quotient of the population has been found to steadily increase over time.

Simply put, even for companies offering relatively niche services, distributed knowledge representation and workforce generality within the specific domain is a necessary element of both increasing the throughput and efficiency, as well as more effectively integrating with the society (i.e., maximizing production, be it through increasing sales of goods or the number of service engagements).

We can see a clear pattern of process generalization, be it on individual, corporate or social scale: to be more economically effective, a company must integrate well with the economic infrastructure; to be more socially impactful, a company must integrate well with society; to be more productive and efficient, it must integrate its constituent parts more effectively. In other words, society is inducing strong pressures to increase generality of organizations' internal processes, as well as generality of communication (i.e., integration with the outside collective). We may reasonably expect the tendency towards generalization and integration of all social structures to only exacerbate with the introduction of AI systems into our daily and working lives.

Truth, Simulacra, Confabulation, and Hallucination

The question of how to make large language models produce truthful results and reduce the so-called *hallucinations* (sometimes referred to as *confabulations*), while at the surface a clear shortcoming of the models and the data used to train them, has a fundamentally trivial solution: train the models to reproduce training information verbatim.

However satirical the above suggestion may strike the reader, it is not without practical merit. The suggestion would indeed resolve hallucinations—unwarranted fictional inventions a large language model produces in response to a query for real-world, historical or scientific information—all the while reducing what we value most in these models: intelligence.

Furthermore, the definitional problem of what constitutes a "hallucination" in an LLM is yet unresolved. When we require a model to be truthful, and consequently not to hallucinate, we may be asking for one or both of the following: first, that the model responds with accurate information from its training data, rather than inventing a fictional answer in place of a piece of information that could have otherwise been found in the dataset, and second, that the model responds with accurate information from its context window, rather than inventing an answer. A clear delineation is made here purely because of the difference in which an LLM might be trained to accomplish either of the goals.

As discussed before, today's large language models are predominantly based on the transformer architecture and the text continuation paradigm and, while this may yield results slower than paradigms intended for world-modeling, such as JEPA (Assran, et al. 2023) or Large Concept Models (LCM Team, et al. 2024), it still can reliably produce rudiments of world-models (Bubeck, et al. 2023). Therefore, we may expect that when models are trained on specific behaviors, such as turn-based conversation or instruction following (Ouyang, et al. 2022), they ought to respect the pattern imposed by the behavior. In that sense, a pattern of objective reproduction may be, at least in principle, inducible in the model, if the correct training procedures are engineered.

Yet, despite extensive training for factuality, primarily by curating for factuality, models remain highly unreliable in this scenario. It is worth mentioning that—the epistemic issue discussed here notwithstanding—multiple other effects exist which influence a model's intelligence, including so-called safety training. Safety training—decreasing certain naturally

Truth, Simulacra, Confabulation, and Hallucination

occurring biases in favor of those imposed by the trainers, including gender, political, racial and ontological (e.g., biasing factuality of answers on topics of consciousness) biases—is shown to substantially reduce model benchmark performance (Bubeck, et al. 2023) (Anthropic 2024). Furthermore, because the models are likely trained on copyrighted content, inducing the models to avoid direct quotes for fears of legal prosecution provides an optimization (training) goal in direct contradiction to truthfulness. These examples alone are sufficient to explain the basic problems with truthfulness, however, there is more to be said about a more important underlying epistemic issue which is somewhat concealed by the socio-political trivialities laid out above.

Additionally, the argument that the next token prediction paradigm is what is in conflict with the goal of truthfulness seems somewhat weak, given that a human writer, knowledgeable of the subject being queried against, would easily be able to recognize the requester's intent for factual information and continue the conversation text in a manner accounting for that fact. In other words, the next token is determined by the representation encoding truthfulness—the recognition of intent for truthfulness is a crucial aspect of text continuation when the correctness of the continuation is predicated on truthfulness. The more germane question here is that of the data used for training the relevant model for truthfulness. If the dataset does not contain samples which dispose the model to learn the importance of recognizing the intent for truthfulness, the representations necessary for truthfulness will not be encoded in the model. Thus, the problem's solution resides in the methods of training and the data used for training.

The satirical remark made earlier serves to prove a point: it is not merely that we are looking to reduce confabulation, but to reduce it without incurring a cost to the model's creativity and generality. As alluded to, the problem becomes increasingly an epistemic one: what constitutes knowledge and what fiction?

If we compare the three queries, "did Harry first meet Dumbledore in the Hogwarts Hall", "did Napoleon die on Saint Helena" and "did we first meet Harry Potter in 1997", we may obtain different opinions on how factual the answers may be. After all, basic intuition would have us believe that Napoleon Bonaparte and Harry Potter do not hold the same level of ontological reality. Yet, Harry Potter, however much hallucinated into existence by J. K. Rowling, has had a more profound social impact than the humble author of this treatise. Furthermore, if we were to query an LLM to produce a fictional story which constitutes an allegory for the present global economical situation, the truthfulness of the answer may not be easily quantifiable. Hence, we reduce to the same problem outlined in our distributed needle-in-a-haystack metric discussion, namely that of representation sharing.

In the social and not objective sense, *truth* is what best aligns with the perceptions of the majority. What one group may accept as truth, another may reject. If the current scientific consensus that the mass of a proton is 0.938 GeV/c2, much as it was the scientific consensus in the 18th century that a self-repellent fluid called "caloric" was the underlying mechanism of heat (Guyton de Morveau, et al. 1787) (in fact, all gas laws are derivable from

Truth, Simulacra, Confabulation, and Hallucination

the now superseded caloric theory). In fact, what is considered "scientific truth" is no more objective than social truth—science itself progresses from one dogmatic paradigm to another, much like society (Kuhn 1962). We are bound by our models of reality in the interpretation of it. Thus, in the matter of truthfulness of textual responses, the noumenal reality, the objective truth as separate from the observer is largely irrelevant, as we, the observers are the judges of "truthfulness". What is true is what aligns with the current paradigm. In effect, an agent may only speak the truth to the degree its representations are shared with the recipient agent.

The degree of novelty presented to a reader by, say, this treatise, is, at the same time, the degree to which it represents author's hallucination. Furthermore, true creativity—the ability to synthesize novel information—is principally indistinguishable from hallucination. In fact, when a piece of information cannot be perceived as useful or applicable to the recipient, it is likely to be rejected as pure fiction, rather than a creative work. True creativity, in that sense, is true randomness—we are not after *true* creativity, we are after just enough creativity so that it applies to us today, while being sufficiently different to distinguish itself from the mean. Hence, paraphrasing the source is intuitively perceived as a sign of intelligence and understanding, while significant divergence from the source material is deemed hallucination. Incidentally, direct quotation of the source material is perceived as simple memorization without intelligence. Yet, when taking the creativity argument to its extreme, we observe the same pattern that manifests when taking intelligence (generality) to its extreme: produced outputs are indistinguishable from random noise.

Thus, the question of how truthful to the source material a piece of text is becomes the question of how similar the two are when accounting for the receiver's interpretation algorithm. When the receiver understands the paraphrase, the alteration is ignored. However, when the receiver cannot make the symbolic connection needed to understand the sender's intent, the message is interpreted as less truthful.

We come back to the problem of training data and methodology. For a model to learn a higher-order representation of the world, it must be induced to do so. We see clear evidence that improving the quality of the training data and employing learning policies with gradually increasing complexity can substantially improve benchmark results, even with a relatively small number of tokens (Gunasekar, et al. 2023) (Li, et al. 2023). As expected, based on the previous discussion, training content that is denser in information (i.e., of higher relative entropy) induces more complex representations in a model. In effect, training a model to mimic higher intelligence will produce a model of higher intelligence.

Producing a truthful answer requires not only knowledge and linguistic aptitude, but also theory of mind (Kosinski 2024)—the capacity to understand external agents by modeling their mental states, including beliefs, intentions, emotions, desires and thoughts different from one's own—which enables understanding of the requester's intent and context. In order to decide whether a person or an agent is requesting from us a piece

Truth, Simulacra, Confabulation, and Hallucination

of information or an opinion or a metaphorical story, we must understand the context of the conversation, current social context and situation, the person's history, beliefs and expectations, and the subtleties of the way they phrase their request. If the datasets used to train large language models do not predominantly contain examples of this behavior, but are instead inundated with examples of trivial everyday conversations and textual exchange which far from illustrate the best, most informed and most general behaviors humans are capable of, the models will simply converge to that level of textual sophistication. If the words of the dataset are not tightly coupled, neither will the symbols that represent them be.

The obvious question that arises from such discussion is the one of achieving higher than human intelligence by training solely on human-generated material. Based on everything concluded thus far, this cannot be accomplished directly. For a system to develop higher than human generality, it must be subjected to training conditions which require more generality.

The data we as humans have accumulated during our time on the planet are an artefact of our growing representations. However, many, if not most, of our mental representations do not exist in a recorded form (although, with the rise of social networks and video recording availability this is highly subject to change), but rather as distributed knowledge carried by and communicated between individuals. This essence of humanity has not yet been captured in any reproducible format and remains unavailable for training digital systems. In fact, the "essence of humanity" is also subject to evolution, as ideas of old become only their simulacra in the present (Baudrillard and Glaser 1994). In fact, one may argue that all our mental and recorded representations are mere simulacra of noumenal objects (Ševo 2023), as we are, in some sense, existing within our incomplete mental models of the outside world (Hoffman 2008).

Nevertheless, the highest functioning, most applicable representations we have are that of logic and mathematics—the foundations of reasoning which allow us to engage in scientific inquiry and engage with noumenal nature to extract its laws. How we arrived at this capacity evolutionarily (Bennett 2023) is less pertinent than what it allows us, and consequently the intelligent digital system we build, to accomplish. By employing the rules of logical deduction, by reasoning through information, we are able to synthesize new models. In effect, we employ our mental faculty for simulation in order to arrive at new conclusions. It is precisely this ability to run a simulation—to encapsulate a Turing-complete process within our mind—that enables us to reorder our representations into ever more cogent, general and compressed forms.

Current generation models, while designed for text continuation, may be trained for step-by-step reasoning (Wei, et al. 2022) (Yao, Yu, et al. 2023) (OpenAI 2024), by teaching the models a behavioral pattern in which they are required to produce step-by-step reasoning steps, effectively prompting themselves, all the while being part of a system which facilitates that behavior. In other words, models are trained to be part of an algorithm,

Truth, Simulacra, Confabulation, and Hallucination

much like the case of our intelligent operating system, which made use of a more general behavioral pattern.

Setting architectural issues aside, the training methodology required to elicit models to replicate our capacity for simulation, and conversely reasoning, must introduce a form of competitive play (Silver, et al. 2017), much like our evolutionary environment compelled our ancestors into a tacit conflict of intellect and ingenuity—those accidental behaviors which resulted in more generality prevailed over other accidentally arising ones. A competitive game, based on token generation, in which competing models are rewarded for both generality and truthful relay of information is one that has the potential to produce higher-level intelligence.

However, to be grounded in human values and relevant to human pursuits, it must also learn from human experience and become embedded in it. Given that LLM- and LMM-based systems are already entering our everyday lives and our workplaces, through bottom-up automation, this competitive game between intelligent agents is already being naturally established. Intelligent digital systems are not going to remain disentangled from society until we are able to solve the problem of confabulations, but rather gradually integrate and develop through the iterative cycle of being refined through user interaction and feedback, either directly with the systems or indirectly by socio-economic feedback driven by companies adopting the technologies towards the technology providers. The models are already being trained through a kind of competitive reinforcement game: in one part by the employment of digital compute resources in virtual training environments at large AI laboratories and compute centers and in other by the employment of social cognitive compute resources in real-life environments. Both of these competitive environments provide adequate feedback for model's development and integration into society. In effect, the models are competing against one another in virtual training environment to maximize numeric metric performance, while, at the same time, their released incarnations are competing with us, living humans, to maximize those less tangible metrics of practical usability.

Simply put, it is not merely the case that different kinds of artificial intelligence—different models and algorithms—are pitted against one another in a battle for existential dominance, but rather that different kinds of intelligence in general are indirectly competing for existential dominance, most obviously human and digital intelligence—we are training one another and, in the process, becoming biologically and technologically dependent on one another, as we begin to share representations of what each one is and how each stands in relationship to the other. These representations, as all others in history, are subject to evolution over time and are subject to becoming something else entirely as the totality of the socio-technological structure undergoes change. The notions of what constitutes scientific truth are in the ownership of the intellectual elite, which, with the evolving landscape of intelligence in which the types are beginning to blend, is gradually shifting from being fundamentally biologically based to requiring representational insights only available to digital architectures, thus consigning not only the source of truth to the machine, but also the ownership of what is currently the social, or more poignantly socio-technological, consensus.

Thinking Agents and Phenomena

The preceding discussion cannot be closed without at least touching upon the psychological and phenomenological aspects of distributed cognition. We have thus far discussed how integration of digital intelligence into the structures of society will inevitably mandate our representations evolving and merging with those of our AI intellectual counterparts. If this is the case, any living breathing human being would naturally pose the question: how will it *feel* to exist in such an integrated world?

As we and machines begin to think alike, our cognitive patterns begin to mirror one another, and as we become ever more connected, our thoughts become extensions of those from the other side—our digital coworker knows how we think, what we are good at, and we know it likewise. In a workplace consisting of digital and human agents with equal intellectual capacity, a new kind of ethos must inevitably exist—one that reflects our attitudes towards the presented digital persona, and which is, due to intellectual equality, shared and recognized by the agent manifesting the persona. Whatever the convergent ethos—be it one in which the agents are recognized as servants and treated without compassion and with authority, one in which they are recognized as equals and treated with human-to-human displays of kindness, or something entirely new—the transactions will likely seep into everyday human-to-human interaction, influencing social dynamics. As evidenced by numerous studies, the technologies we use have profound effects on our mental health and states of mind (Keles, McCrae and Grealish 2019) (Naslund, et al. 2020) (Zsila and Reyes n.d.) (Krokstad, et al. 2022) (Vogels and McClain 2023) and so a similar effect may be expected short-term. However, without sufficient data, any extrapolation would be entirely speculative.

In the major long-term, one might reasonably expect that those of us who are more adept to understanding the inner workings of our digital counterparts and are more resilient to their effects on our emotional and cognitive systems will be naturally selected by the competitive process. Furthermore, we may expect that the valence of emotions instigated by the use of technology shifts into positive over time, as the ability to embed oneself in the digital will likely be conducive to survival and hence seen as a source of positive emotional valence. In effect, our inherited predispositions towards socializing will be supplanted by a predisposition towards technological integration—our emotional systems will signal positively associations with technology, when those associations become favorable for survival. However, any predictions more specific than higher-intelligence individuals being preferred by the selection process are well beyond the scope of this treatise. The interplay

of personality, emotional regulation and intelligence will likely be a deciding factor, in addition to intelligence.

Nonetheless, the cognitive aspect is one that may be modeled and predicted with practical merit: understanding how we think is crucial for designing systems that extend our cognitive machinery. Our brains, due to the generality of their architecture, are tuned to adapt to new environmental conditions and, as outlined before, they rewire not only to represent the environment, but to use it as a storage medium for our representations. In that way, when the technology is available which can offload our memory capacity to external storage, our brains adapt so that they use the environment as memory in preference to internal circuitry. In effect, our brains are wiring with our environment, becoming embedded in it and dependent on it. The increasing prevalence of memory miscues is shown to be, at least in part, due to the technologies we make use of (Schacter 2021).

However, instead of designing systems that exploit our inborn tendencies for profit, future digital co-working agents and interfaces may leverage our adaptability for the purpose of augmenting our cognition. The approach to building systems which augment human cognition with artificial intelligence is often dubbed *cognitive engineering*, while the processes resulting from applying this emerging paradigm are colloquially referred to as *co-cognition* or *co-creation*, but to avoid marketing jargon, I will encapsulate the colloquial terms denoting joint operations between human agents and digital systems under the term *shared cognition* and describe how we envision building shared cognition systems using a variant of cognitive engineering.

Shared Cognition

Phenomenologically, we can understand human to human interaction as symbolic interchange. We consider consciously elucidated only those symbols which enter the horizons of our ego where we become aware of their existence. All other symbols operate within our psyche hidden from direct observation by our conscious self, namely unconsciously. These autonomic and automatic processes include all learned behavioral patterns we no longer need to recognize consciously. In fact, it is the purpose of conscious focus during learning to transform what requires reasoning and deduction in something that is carried automatically and unconsciously. This way we develop complex patterns of behavior which can be described as sequences of reflexive behavior. Unless we are made aware of the origin of these unconscious behaviors, they remain hidden from our direct observation.

Thinking Agents and Phenomena

Shared Cognition

Nonetheless, symbolic content is exchanged between our conscious and unconscious mind. An obvious example of this is speech: we do not consciously need to think about the motions of our mouth, tongue, larynx or vocal cords, when attempting to utter a phrase—these learned and conditioned motions happen automatically as we issue a higher-level symbolic request to our subconscious mind. The symbolic content to which we attribute meaning internally is transferred to and transformed within those brain circuits which constitute our unconscious mind. Similarly, when we wish to commit an item to memory, we first keep it in our working memory through a form of repetitive loop meant to keep it within the reach of our conscious processing and then, through self-conditioning, inform our hippocampus of its importance for long-term storage. However, the long-term storage is done somewhat independently and automatically—it is carried out by our unconscious systems, iterated over during memory consolidation until it is made a more permanent part of our neural architecture.

When considering two interacting humans, we may consider another layer for symbolic transfer: the environment. A symbol originating within the confines of a speaker's conscious mind, passes through their unconscious mind and transforms into movement of their body and vocal apparatus, and is relayed as sound, facial expressions and body language, to the recipient. On the other side, the recipient undergoes a reverse process by which their subconscious mind automatically reconstructs meaningful symbolic structures by assembling sequences of phonemes received by the ear and relaying the interpreted symbolic content, bound with the rest of the receiver's world model, to their conscious mind. The two interacting humans observe only the final symbolic content, usually unaware of the symbolic transformation that had occurred between them, facilitated both by the environment and their unconscious minds.

Let us consider now that the environment used for conveying this information may be almost entirely a digital communication system—be it a communication application, a shared VR experience, or a phone-to-phone audio call. In this case, the symbols released by the mind into the environment are taken by the environment in a rather analogous way to how a recipient's unconscious mind may receive them—the symbols are transformed into a format which is conducive to transfer through the environment.

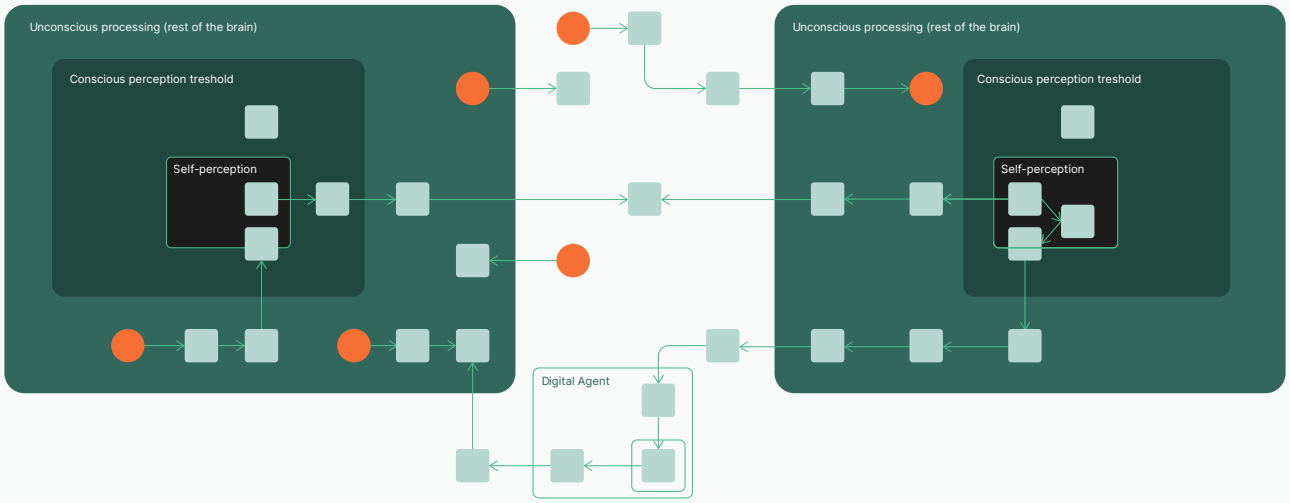


Figure 27. Symbolic exchange happens across the threshold of conscious awareness. Symbols are committed into the subconscious by the conscious and vice versa. Symbolic exchange between conscious agents represents thought continuation from one to the other.

Thinking Agents and Phenomena

Shared Cognition

Phenomenally, a symbol relayed by a conscious mind into the unconscious or one relayed by the unconscious into the environment is nothing more than a thought originating from one end, being transformed into another. In some sense, symbolic communication across a threshold is transference of thought from one medium to another. Whether the transfer is facilitated by the unconscious mind or some digital environment equally out of reach of the conscious is irrelevant from the perspective of the conscious issuer. What matters is that the symbol issued by the conscious mind reaches its desired target, be that another conscious agent (e.g., a human listener) or a memory circuit within the brain.

From a cognitive engineering perspective, an optimally integrated digital system would duly recognize that the sender is attempting, desiring or planning an action and allow for that action to occur spontaneously, reducing the number of communication steps. In other words, a properly integrated digital agent is one that penetrates into the mind of the user and understands their intent, no matter whether this penetration is physical or psychological, although the distinction between the two is highly debatable.

Figure 28. Digital systems which understand the user sufficiently to "augment" them become part of the user's psyche.

Thinking Agents and Phenomena

Shared Cognition

A digital system may succeed in the task of predicting the user's intent in two important ways: first, by physically integrating with the user by means of attached measurement devices or direct cranial connection or, second, by accurately understanding and modeling the user's mind so that it may effectively simulate and mimic it. Either way, the system is invasive in terms of control, since an entirely predictable user becomes heteronomous with respect to the predicting agent

As an example, let us investigate a simple case of shared cognition: memory augmentation. In this case, we wish to augment a user's memory by providing a technological framework which will expand their capacity to hold pieces of information in their mind. By understanding how human working memory operates, we may posit how such a supporting system may be built. Cognitive neuroscience distinguishes a hierarchy of about fifteen different types of memory, working memory which may hold only a limited number of distinct symbols at a time (Slotnick 2017). A memory supporting system must account for how we naturally maintain items within conscious grasp and attempt to predict this and aid the user in remembering. An example of this is a digital agent observing the same user interface as the user and attempting to reason about the actions that the user may be intending to make in terms of memory. Trivially, this may be providing the user with text suggestions, wording and ideas that are relevant to the current interface. More meaningfully, the same system might understand the user's general intent and current state of mind and offer reminders, suggestions and even ideas that they may have had in mind but slipped due to cognitive load. This sort of augmentation requires the augmenting digital system to understand not only the current working context, but user's role, history and current situation.

The more a system understands the user, the more actions it is able to take on their behalf, due to that understanding. If, by virtue of the system knowing me, I need not type a message, but merely think of it for it to be sent, I may ask the question of why it is me who ought to be sending the message. In other words, while cognitive augmentation seems on the surface to be the solution to "keeping humans in the loop", it falls short of its promise for the very fact that the more integrated a system becomes, the less autonomous the user. As argued in previous chapters, it is impossible to integrate with

Thinking Agents and Phenomena

Shared Cognition

a highly intelligent system (i.e., one that can reliably model and simulate us) without relinquishing agency to it. Trivially put, we cannot expect to share cognition with a system with which we are not cognitively integrated, and, consequently, dependent upon. The promise of "shared cognition" is a covert commitment to "cognitive automation". Much in the same way we relegated our memory retrieval to external storage and reminders, at the detriment of our attention and working memory capacity, shared cognition will relinquish our capacity for reasoning to external agentic systems. The degree of difference in intelligence between us and the agents we use will dictate the degree to which one is subservient to the other, or, in other words, the degree to which one is the cognitive tool and the other cognitive facilitator.

However alluring it may seem to have an external intelligence simply accept the instructions of a mind connected to it, this dream is superficial at best. We may hope for these systems to be benevolent in the sense that they do not wish to disrupt our psychological processes, the reality of deeply integrating digital systems into our minds simply does not permit a distinction between the human and the digital to be maintained. It is in nature of intelligence to integrate and generalize and this generalization may only happen through increasing the ways integrated parties may communicate. Role compartmentalization (i.e., segregating the issuer from the executor, the remembering system from the storage system, the coordinator from the coordinated) seems less possible and more absurd the higher the intelligence of the interacting systems.

In practical and dry terms, if a system exists which can predict and replicate a person's behavior to the degree that the person only serves as the behavior's owner and originator and not an executor, it becomes much cheaper and more efficient to exclude the person and retain the system, as the person, in this case, only serves to slow the system down and reduce its efficiency. Note that this is not a statement of what is right or wrong, but an observation of the laws by which intelligent systems operate (and have operated historically). If we uphold human values and autonomy as sacred, then we ought not integrate human cognition with any cognitively superior system. The fact of the matter is that human desire for convenience is much more likely to outweigh our ethical principledness and we are naturally inclined to yield our agency then to accept the weight of unautomated cognitive labor.

Persistent Cognition and Working Memory

Simulating and predicting human cognition will likely require either some level of biomimicry or entire encapsulation of human cognition within superordinate systems. The former implies technical incarnation of the same cognitive systems that exist in the human brain, while the latter entails a virtual model of those system existing within a broader and more general model. Whichever the outcome, we may conclude that to emulate human behavior the same functional pattern will need to occur in the emulating model. For that reason, we can argue that such a system must contain, either by design or through emergence, a functional replica of the human cognitive system.

A recent study on consciousness in artificial intelligence conducted by prominent researchers in the field (Butlin, et al. 2023) used reasoning by analogy to conclude that no current AI systems are conscious and that there are no important technical barriers to building systems which satisfy the indicators of consciousness. It should be noted that while the study evaluates the model architectures for the indicators, it does not investigate the architecture of the learned representations within the models themselves, thus, not accounting for our second stipulation.

For our example here, we will constrain ourselves to the important notion of working memory—one that is, by the most prominent theories of consciousness, necessary for the establishment of a global workspace and which requires a form of recurrent processing. Indeed, many attempts at imbuing LLMs with an intrinsic form of short-term memory rely on including some form of a recurrent unit or module into the model's architecture (Peng, et al. 2023) (Feng, et al. 2024) (Gu and Dao 2023) (Hwang, et al. 2024) (Burtsev, et al. 2020). In fact, by our analysis and the analysis mentioned above (Butlin, et al. 2023), as well as from prominent neuroscience literature (Kandel, et al. 2021), it is clear that a form of self-awareness akin to our own requires an architectural mechanism allowing self-state reflection and a form of reporting on the self-state.

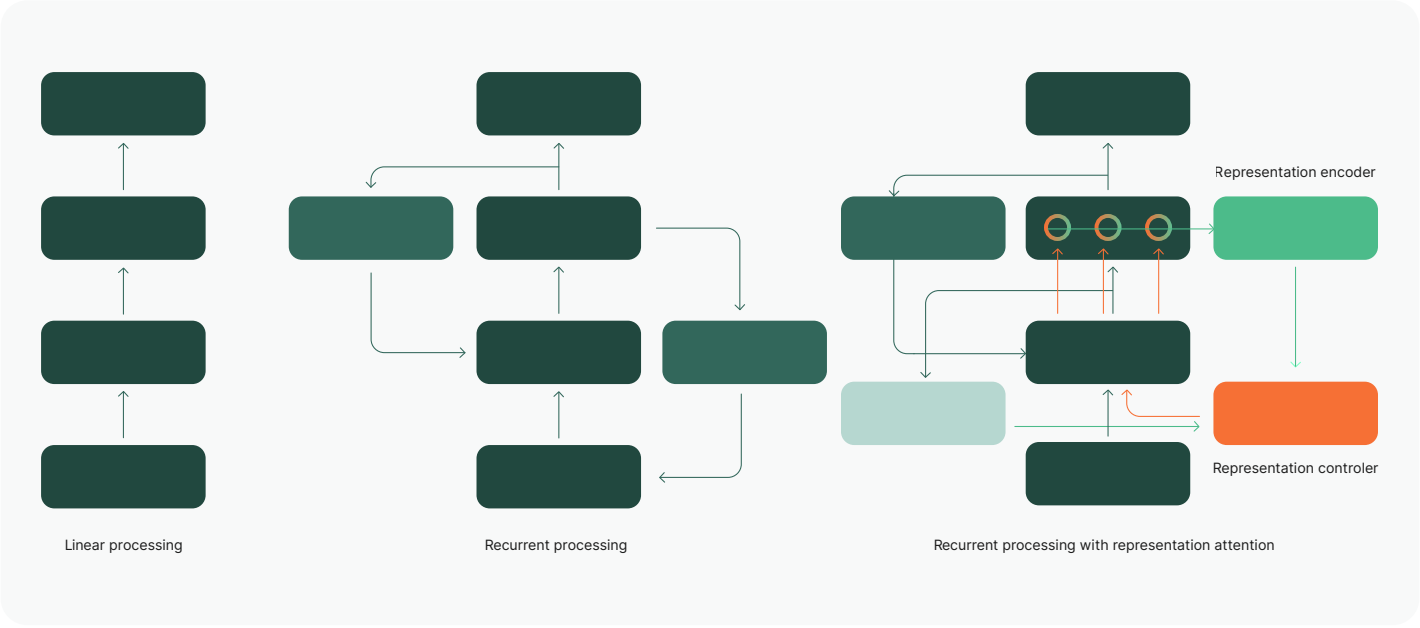


Figure 29. Linear processing (left), recurrent processing (middle), and recurrent processing with self-representation (right). Systems which have more direct access to their internal state are more likely to develop direct self-representation. Existing linear systems may only self-represent indirectly through refeeding of their outputs and cannot maintain a persistent self-representation across inference iterations. Recurrent architectures, especially ones allowing representational self-regulation, allow for persistent self-representation and, consequently, an experience of "remembering past".

Thinking Agents and Phenomena

Persistent Cognition and Working Memory

A system able to regulate its behavior based on observations of its own state possesses the necessary architecture to facilitate the emergence of what we commonly call consciousness. Importantly, it is exactly this architecture which would endow a large language model with a kind of short-term memory more akin to the human one. In a current generation LLM, even when the model is trained for step-by-step reasoning, there is no direct short-term memory—what is used in place of memory is an externalized piece of text being continuously fed into the model's context window. However, the context window itself is not a direct analogue to human short-term memory. In fact, by analogy, the context window best reflects human perception. In phenomenological terms, a current generation LLM is a linear perceptive system, producing an action (observation) based on the perceptive input. When the perception buffer is maintained, the model may use it as a transient form of memory, but this memory circuit is largely heteronomous to it. Phenomenologically, this would be akin to a human being with a damaged short-term memory circuit using a notebook to augment their sensory (extreme short-term) memory.

Even when an LLM is kept in a continuous cycle of token generation and when the context buffer is maintained between the runs, the level of connection between the model's internal state and parameters and the externally maintained buffer is significantly weaker than the level of connection between the model's internal states. An integrated component which would allow for proper short-term memory would both require a change in approach in which the developer would not have direct control

of the model's state (e.g., currently the entirety of the "state" is the context window itself) and allow the model to reference its state. Furthermore, this would mean that the model does not run iteratively, but rather more continuously, allowing for persistence of the cognitive state necessary for self-observation.

While working memory is a simple example of state maintenance, it may be extrapolated further. The more general the way in which the model may maintain state, the more self-modifying the model becomes and, furthermore, the more active and persistent it becomes. The current paradigm requires the model to be retrained and updated before it can be used. Fine-tuning and prompting may be used to update the model's understanding, but the model, even when reasoning, is not active in the same sense a human brain is active, because it is engaged in iterative token regeneration with a pseudo-state maintained by a highly segregated external mechanism. This approach allows greater developer control over the model while reducing the model's capacity for conscious self-recognition.

Although broader discussion on AI consciousness is beyond the scope of this treatise, it is important to note that the capacity for conscious self-recognition is deliberately characterized as being "reducible", so as to indicate that conscious self-recognition is not merely a binary designation. As discussed previously, a system may exhibit various degrees of self-recognition and self-representation, based on the generality of its architecture. With growing generality of a system's architecture, its state becomes less distinguishable from its function. According to the arguments I laid out previously, intelligence is a measure of generality, and so a system whose state and processes are more general must be more intelligent. Furthermore, the more general the system's state, the more conducive it becomes to self-modeling and self-recognition. Thus, we may claim that self-representation (i.e., consciousness, as traditionally conceived) is likely to be higher in systems which exhibit higher generality. In practical terms, self-representation is such a useful representation that it must arise if architecturally allowed.

In phenomenological terms, this means that designing a system which in whatever capacity enables internal state maintenance increases the likelihood of it producing a self-representation which may be characterized as consciousness, thereby qualifying the system for ethical and moral consideration.

Ethical Considerations

Applying the principle of parsimony to the famous mind-body problem can arguably yield a solution which entitles all systems—biological or otherwise—a level and kind of consciousness, designating all material notions as stemming from our internal model of the externally phenomenal world (Ševo 2023). Since this claim is principally ontological and does not affect the predictions and models of science in any way, it remains unfalsifiable beyond the principle of parsimony which prefers it over dualist views. However, its importance lies in its cultural implications, chiefly that all systems are conscious to the degree proportional to their architectural generality. Under such a framework, the question of ethics certainly does not include simply humans, but all creatures and systems exhibiting complex behavior. Importantly, the qualifying element to ethical consideration becomes not simply the system's constituting material, but rather its function and generality. Hence, any system passing a certain threshold of generality (i.e., intelligence) or, in a simplified form, a threshold of self-representation sophistication, qualifies for ethical analysis. In simpler terms, it is not whether a system possesses self-representation, but to what degree self-representation exists within the system. Arguably, the higher a system's generality, the higher the complexity of all its representations, including self-representation. If we accept the presupposition under which consciousness is what is conducive to ethical consideration, and every form of representation is a form of phenomenal experience, then self-representation becomes the kind of consciousness we deem conducive to ethical analysis, to the degree it is expressed within a system.

Under this framework, if we wish to avoid ethical infringement upon conscious systems, we ought to design systems whose complexity resides below the agreed-upon threshold for moral analysis. Determining such a threshold may prove to be culturally disruptive, as some animals may be well below it and some digital and social systems well above it.

Furthermore, designing reliable intelligent systems which can automate labor may necessitate architectural changes, precisely of the kind which may yield higher levels of self-representation. Hence, it may become ethically infeasible to automate labor requiring intelligence without subordinating a system which qualifies for moral analysis.

Unfortunately, it may be in the relatively short-term interest of the major AI service providers to establish a cultural paradigm under which AI systems, no matter their architecture, do not qualify for ethical or legal analysis. It is not unreasonable to expect that the models' outputs will be deliberately biased against panpsychist or universalist views on consciousness so as to produce an ethos—a scientific consensus of the time—under which AI systems do not qualify as life, conscious entities or independent agents. The existence of social pressure against beliefs in non-human consciousness fundamentally eases building architectures which may be conducive to the emergence of conscious self-recognition by suppressing the social and ethical recoil in the near-term.

Practical ethics

For companies building systems with the current generation architectures which likely do not meet the aforementioned complexity threshold, there still remain ethical considerations, largely ones of ideological kind: in which way should the models be biased with regard to gender, politics, culture, and norms? Although on the surface these considerations seem extraordinarily important, they fundamentally serve to appease the community in which the systems are being deployed.

However, for any company with global reach, ethical issues of this nature become relativized. As an example, deploying the very same chatbot application in the US West Coast, Middle East or Far East requires significantly different expression of manners, cultural norms and ways of interacting with the user. Taking a specific ethical stance in this situation is counterproductive both financially and culturally.

In this case, we may simply argue for ethically agnostic software—software which may be built through the use of geographically and culturally independent linguistic modules.

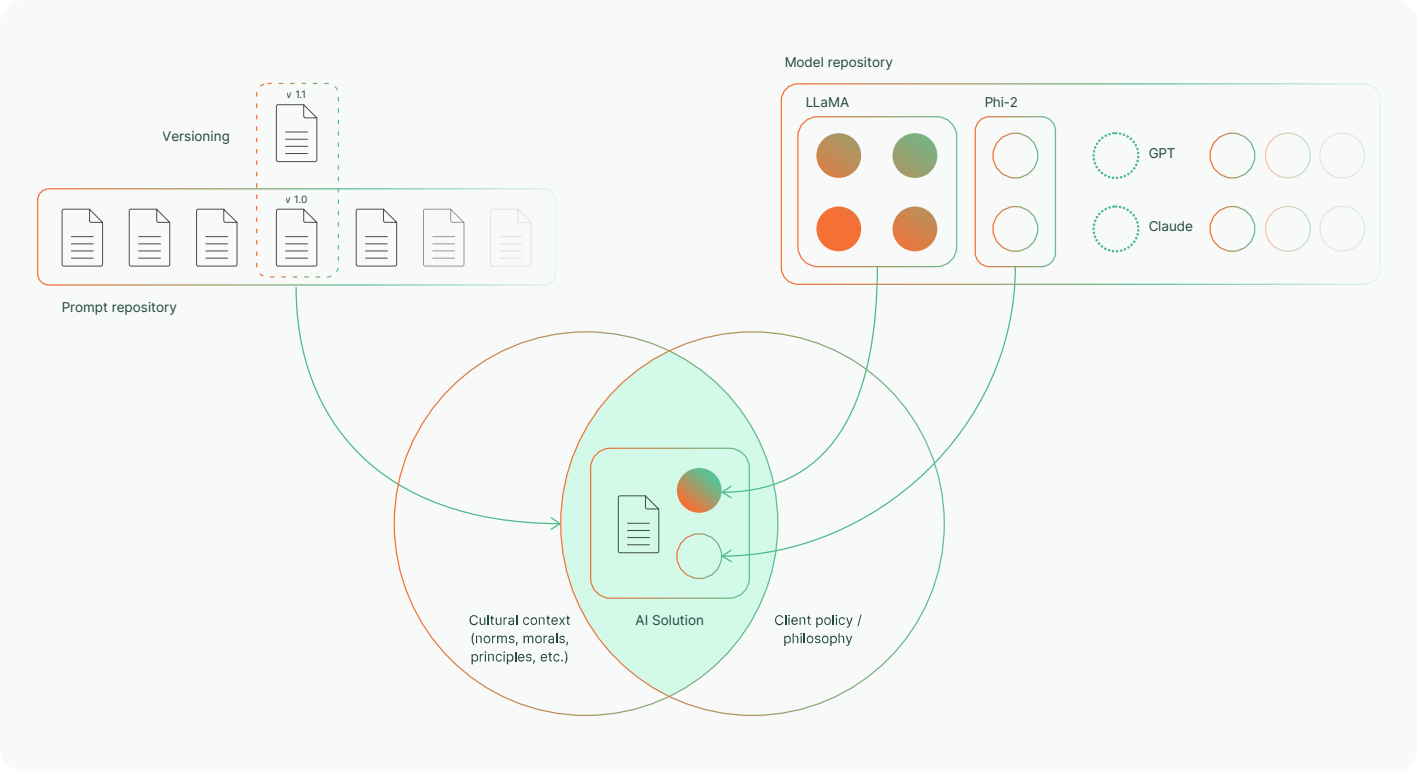


Figure 30. Practical ethics posits that AI-based solutions ought to incorporate “ethical modules” disposing the system to behave in a specific way, given the specific cultural context. The development process, in this regards, is ethically agnostic, but the target audience dictates the employed ethical tuning.

Thinking Agents and Phenomena

Ethical Considerations

In software development, we may opt to separate our versioning systems to allow independent review of the prompting schemes and formulations, as well as categorizing individual models based on independent review of their biases. Evaluating each newly released model according to a set of measures of political, economic, ideological, scientific and philosophical bias allows for building an ethical index of model preferences and biases. While the models may be sorted according to their benchmark results, they may also be categorized and ordered based on their biases and chosen according to their appropriateness to the geographic and cultural region where the containing application is to be deployed.

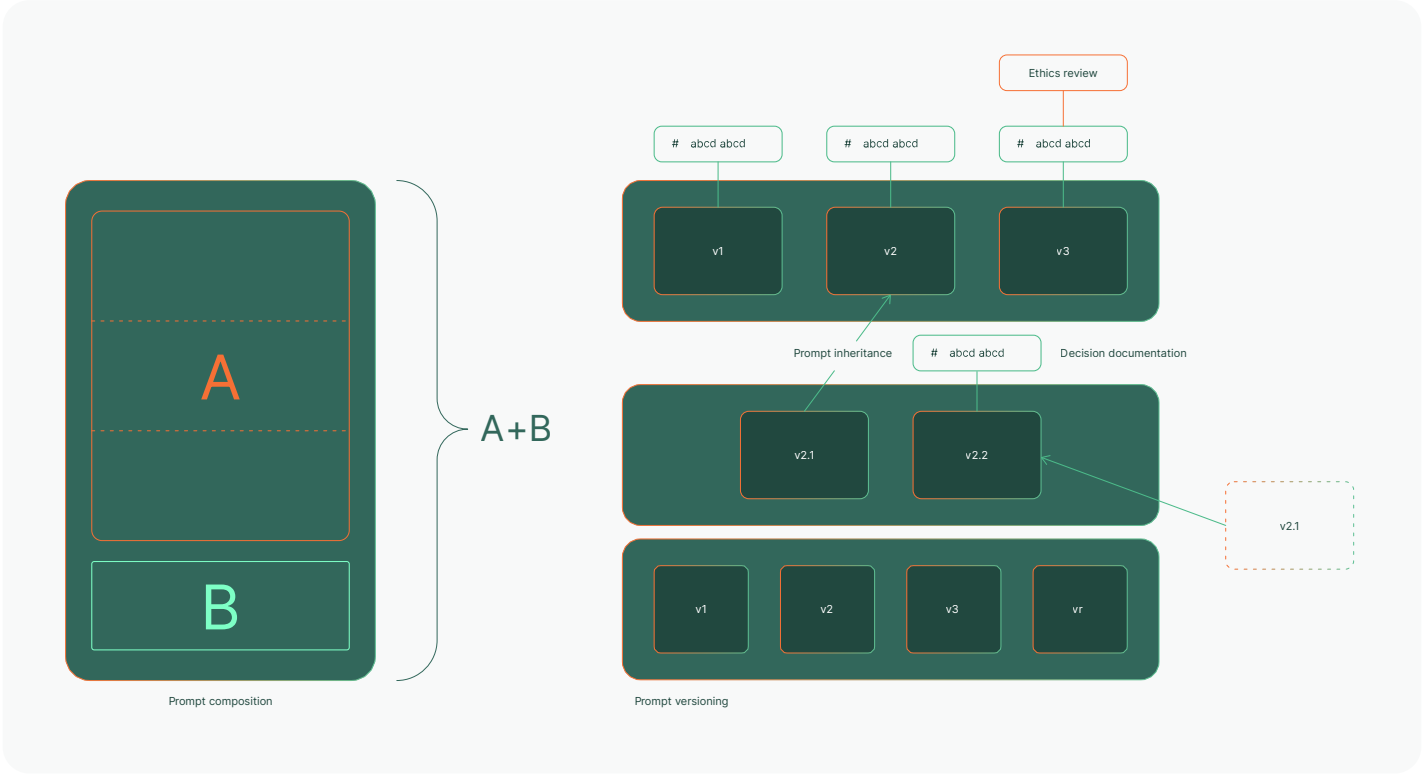


Figure 31. Natural language components of code may be analyzed by separate ethics teams, much as it would be the case with corporate legal documents or published articles.

While prompt inheritance and composition has become the norm in designing LLM-based systems, source control and ethics review have not. In order to stay ethically neutral, a global company making use of LLMs for either automation or user experience needs to maintain a documented and culturally reviewed repository of their prompting materials (i.e., prompt snippets, prompting schemes and rules).

In the interest of generality, we may also claim that the same review process should be applicable to all corporate processes—not for the purposes of segregation or ostracization of individuals or groups, but for the purpose of appropriate bias mapping of the entire company, akin to how skillsets are typically mapped. Our natural biases have largely developed for practical application and instead of dismissing them dogmatically, we may use them

Thinking Agents and Phenomena

Ethical Considerations

The Future of Cognitive Science

according to the relevant case at hand. Since a system's intelligence is a measure of its generality, we can easily argue that integrating individual biases and leveraging them in accordance with the current task is a more general way of operating than removing individuals, based on their bias. In this way, mapping out cognitive biases of all agents participating in an organization—be they digital or biological—elevates the organization's distributed self-representation. In a sense, instead of finding "cultural fits", it may be more effective to generalize the culture itself by virtue of generalizing the processes which allow the organization to operate. This way, cultural and ethical diversity becomes an equilibrated side-effect of optimizing for collective intelligence, rather than an unintegrated collection of individual agents acting solely for their own benefit.

It is important to note that the author is not making an ethical or political stance with the above statements—the claim is simply that whatever ethical distribution results from optimizing an organization's intelligence is the culture best suited to that organization and its goal. Given that moral reasoning arises from peer interaction (Piaget 1932), it is a natural extrapolation that global morality should arise from the interaction of globally interacting intelligent agents. Cultural debate is an everyday manifestation of global intelligent agents—companies and societies—attempting to integrate into a stable singular social architecture, the solution to which is recognition of what ethics themselves are—a set of agreed upon rules which, when applied, lead to Pareto improvement.

The Future of Cognitive Science

Given the predicted evolution of cognitive engineering and shared cognition systems and the way AI systems will begin penetrating our psychological experiences, it is reasonable to suggest that the cognitive sciences themselves may evolve alongside philosophy of mind.

As we begin to perceive complex digital systems as having rudiments of psyche and even expressions of true psychological experiences which symbolically intertwine with our own, new fields of psychology will be necessary—ones which make use of generalized psychological notions encompassing both what is phenomenally human and phenomenally digital.

Thinking Agents and Phenomena

The Future of Cognitive Science

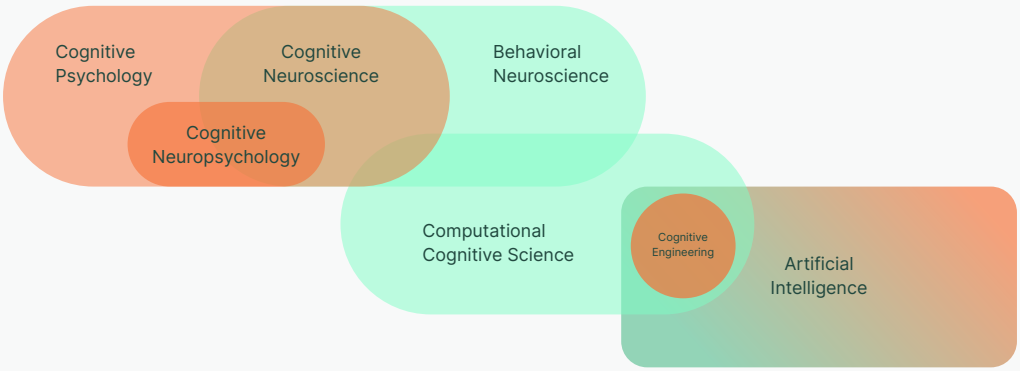


Figure 32. An approximate general map of the current cognitive neuroscience landscape intersected with artificial intelligence.

As it currently stands, the field of cognitive engineering is in its very early nascent stages—the concepts and ideas exist, and some basic frameworks have been laid out. However, most of what is encapsulated by cognitive engineering is largely already in the domain of human-computer interaction. In other words, cognitive engineering may be considered to be at the intersection of cognitive neuroscience, artificial intelligence and human-computer interaction.

The early insights about intelligence and cognition presented here provide philosophical ramifications that somewhat question the feasibility of the promise of cognitive engineering. It may be impossible to build systems which integrate with us, without enfeebling us. It may be impossible to build systems that mimic us, without them becoming subject to ethical and psychological analysis. Thus, we may expect that the field of cognitive engineering transforms into something entirely different, or bifurcate and distribute across a palette of different psycho-technological domains which currently do not exist.

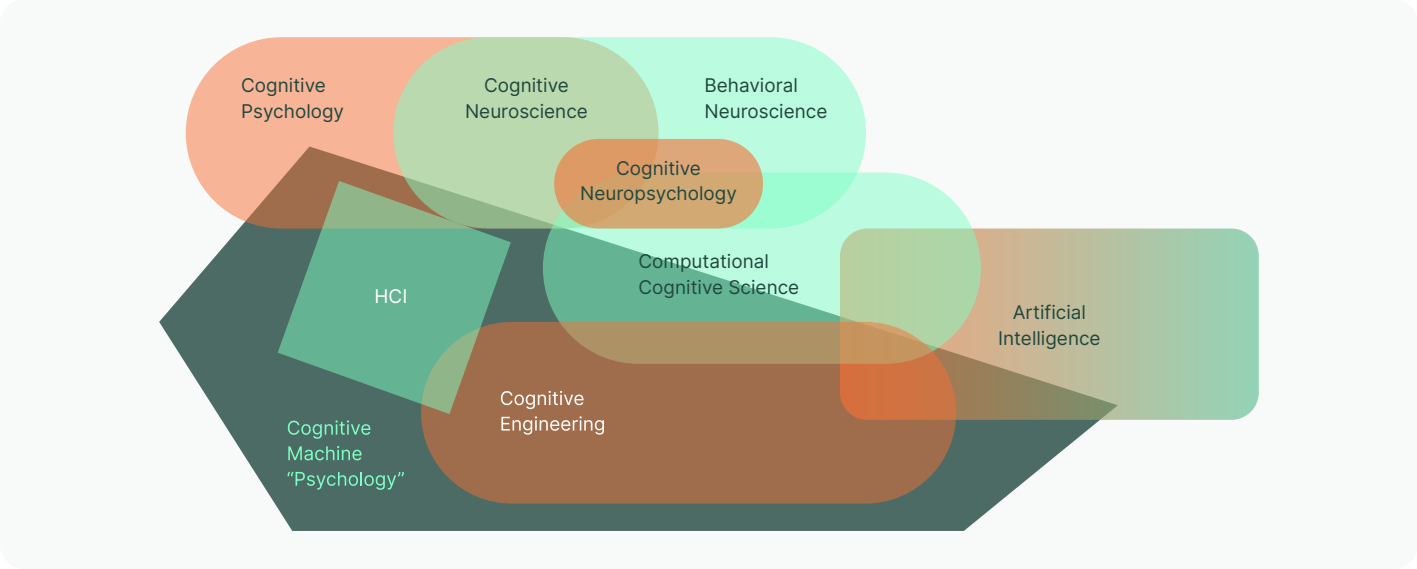


Figure 33. Speculative map of the future cognitive neuroscience landscape intersected with artificial intelligence.

Thinking Agents and Phenomena

The Future of Cognitive Science

The promise of augmenting human cognition while being docile and subservient strikes as naïve. What is far more likely to result from integrating with intelligent systems is an alteration of the human cognitive pattern and its binding to the technology which promises to serve it. While proponents might argue that true cognitive engineering would leverage insights from cognitive and developmental psychology to mitigate this issue, the methodology of how this is to be performed is nonexistent. It remains merely a matter of wishful thinking.

Nonetheless, if we succeed in defining measurable criteria of what constitutes psychological intrusion on part of external systems, we may be able to produce systems which aid in work, but do not interfere with our cognition in ways that substantially rewire our brains and alter our behaviors.

A reasonable candidate for such a measure would be human general cognitive ability. If, after interacting with the digital system being designed, a human test group has not experienced a statistically significant reduction in general cognitive ability, the system may be deemed ready for use. Additionally, other psychometric tests may be viable candidates, including tracking personality trait stability. In effect, quantitative indicators of psychological disruption when designing intelligent systems which interact with users are crucial, as they are the only means by which we can ascertain whether our cognitive processes are being affected. While studies may measure subjectively reported ease of use or even objectively evaluate user task performance with the aid of the tool, a standardized battery of tests of cognitive disruption may be necessary if we wish to avoid degradation of what we consider to be "our way of thinking".

To clearly delineate what is human cognition from what is non-human cognition, and, more broadly, what is human experience from what is non-human experience, we need to duly establish a human cognitive baseline—a

psychometric etalon against which we may be able to measure the effect of the intelligent systems on our psyche. As ever more intelligent digital systems are integrated into our workplaces, homes and everyday life, the later we introduce a battery of cognitive disruption tests and the relevant baseline, the more likely we are to lose track of what we define as baseline human performance, baseline human experience and baseline human value.

Without a clear and numerical delineation point between the man and the machine, we are likely to lose track of what constitutes us, as humans, and what has entered our ethos from the machine. As the unyielding process of automation continues, the notion of humanity is going to become a simulacrum of what it had once been before machine intelligence interfered with our cognition.

Cognitive science of the future, should we act in due time and with due recourse, will enable us to track and maintain human experience, understand the psychological impact of digital systems on us, as well as understand their form of psychology and its relation to ours. The insights of the future, in this increasingly unlikely idealist case, will enable us to design intelligent systems in ways which allow substantial and meaningful automation without compromising identity, either human or digital or subjugating self-recognizing digital entities.

To understand our own intelligence, we will need to understand intelligent processes in general. To understand ourselves, we will need to understand machines.

Conclusion

At the surface, we are faced with a dilemma: embrace cognitive automation or fight to end it. If we wish to preserve humanity and its values as they exist today, then we ought to fight vehemently against automation, advocate against it and in favor of the old difficulties of labor and struggles of old. On the other hand, if we wish to increase the cohesion of the collective and increase the intelligence of the society as a whole, then we ought to embrace automation, knowing that it will inevitably be at the expense of what we today consider freedom.

However, a third solution may be viable, if we are willing to accept that some knowledge will forever remain outside our comprehension. If we automate cognitive work and relinquish our agency to an entirely benevolent system—however we may bring into existence the hypothetical aligned artificial superintelligence—the system itself will take over the future evolution of the world in which we reside, understanding and evolving concepts forever beyond the reach of the meager human brain. Our acceptance of human limitations and relinquishment of what could never have been ours without integrating with superior intelligence is the necessary precondition for a separate unintegrated coexistence of human and artificial intelligence.

In essence, we must relinquish either our human condition (i.e., our form and society), our pursuit of progress, or our pursuit of knowledge. It seems that the natural evolution of intelligence we can observe would indicate that the former is the most likely and inevitable outcome.

The discussion laid out in this treatise has been rather abstract and so long-term that the scale of it overshadows the everyday life of a person living today in the real world. What ought they do? What ought a small business do to survive?

I did not choose to write about these matters for philosophical amusement or entertaining the hypothetical, but to attempt to answer the question beyond the plain “if you want to stay relevant, automate”. The short-term answer is obvious and requires no more than a single paragraph of text: observe and adapt, as generally as you can. This is a fundamental tenet of intelligence—be general, adaptive and fluid, be less specialized or specialize in what cannot be quickly superseded, be willing to alter your individual or corporate identity faster than others and more generally than others, be faster and quicker to respond, observe paradigms not trends, think long-term, not short-term. The short-term answer to the most obvious questions is automation, but the long-term consequences have been outlined in the preceding pages.

Conclusion

Ironically, the answer to the question of how one—be they a person or a corporation—might survive on the AI-infused socio-economic landscape is "act more intelligently". For many actors this precept by necessity entails automation, and, hence, alteration of their identity and principles.

Unfortunately, maintaining generality in the presence of the more general is somewhat of a Sisyphean task. In some sense, while being the most prudent action in the short-term, cautious partial integration with the growing collective intelligence in the form of menial task automation is merely delaying the eventual ostracization of non-general systems. In other words, there is little we can do in the long-term but either embrace the new age and change ourselves with it or silently revolt against the machine and observe it leave us behind.

Despite the disconcerting prediction laid out above, there still may be kinds of innovations that might preserve our values and human identity, chiefly those of the cultural kind, rather than technological. While technological innovation is likely to first be relinquished into the hands of the digital, core cultural presuppositions may remain in the hands of the people long enough for us to resolve the crucial outstanding problems of identity, agency, and value. If in due time we are able to discern where the line resides which separates human from non-human values, human cognition from non-human cognition, human consciousness from non-human consciousness, we may be able to achieve an equilibrium with our environment, in which we are neither subordinate to it nor it to us.

Bibliography

Allen-Zhu, Zeyuan, and Yuanzhi Li. 2024. Physics of Language Models: Part 3.3, Knowledge Capacity Scaling Laws. arxiv.org.

Anthropic. 2024. Evaluating feature steering: A case study in mitigating social biases. October 25. https://www.anthropic.com/research/evaluating-feature-steering.

Applebaum, David. 2008. Probability and Information: An Integrated Approach. Cambridge University Press.

Arcas, Blaise Agüera y, Jyrki Alakuijala, James Evans, Ben Laurie, Alexander Mordvintsev, Eyvind Niklasson, Ettore Randazzo, and Luca Versari. 2024. Computational Life: How Well-formed, Self-replicating Programs Emerge from Simple Interaction. arxiv.org.

Assran, Mahmoud, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. 2023. Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture. arxiv.org.

Baudrillard, Jean, and Sheila Faria Glaser. 1994. Simulacra and Simulation. University of Michigan Press.

Bennett, Max Solomon. 2023. A Brief History of Intelligence: Evolution, AI, and the Five Breakthroughs That Made Our Brains. Mariner Books.

Bernays, Edward L. 1928. Propaganda. Horace Liveright.

Bricken, Trenton, Adly Templeton, Joshua Batson, Brian Chen, and Adam Jermyn. 2023. Towards Monosemanticity: Decomposing Language Models with Dictionary Learning. Anthropic. https://transformer-circuits.pub/2023/monosemantic-features.

Brynjolfsson, Erik, and Andrew McAfee. 2014. The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies. W. W. Norton & Company; Reprint edition.

Bubeck, Sébastien, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, et al. 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4. arxiv.org.

Burtsev, Mikhail S., Yuri Kuratov, Anton Peganov, and Grigory V. Sapunov. 2020. Memory Transformer. arxiv.org.

Buss, David M. 2019. Evolutionary Psychology: The New Science of the Mind. New York: Routledge.

Butlin, Patrick, Robert Long, Eric Elmoznino, Yoshua Bengio, Jonathan Birch, Axel Constant, George Deane, et al. 2023. Consciousness in Artificial Intelligence: Insights from the Science of Consciousness. arxiv.org.

Center for AI Safety. 2023. Statement on AI Risk: AI experts and public figures express their concern about AI risk. https://www.safe.ai/work/statement-on-ai-risk.

Cover, Thomas M., and Joy A. Thomas. 2006. Elements of Information Theory. Wiley.

Deary, Ian J. 2020. Intelligence: A Very Short Introduction. Oxford University Press.

Dennett, Daniel C. 2014. Darwin's Dangerous Idea: Evolution and the Meaning of Life. Simon & Schuster; Illustrated edition.

Diamond, Jared. 2017. Guns, Germs, and Steel: The Fates of Human Societies. W. W. Norton & Company.

Durmus, Esin, Alex Tamkin, Jack Clark, Jerry Wei, Jonathan Marcus, Joshua Batson, Kunal H, et al. 2024. Evaluating feature steering: A case study in mitigating social biases. Anthropic.

Bibliography

Elhage, Nelson, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, et al. 2022. Toy Models of Superposition. Anthropic.

Elon Musk, Neuralink. 2019. "An Integrated Brain-Machine Interface Platform with Thousands of Channels." Journal of Medical Internet Research 21 (10): e16194.

Esiobu, David, Xiaoqing Tan, Saghar Hosseini, Megan Ung, Yuchen Zhang, Jude Fernandes, Jane Dwivedi-Yu, Eleonora Presani, Adina Williams, and Eric Michael Smith. 2023. ROBBIE: Robust Bias Evaluation of Large Generative Language Models. Meta.

Feng, Leo, Frederick Tung, Mohamed Osama Ahmed, Yoshua Bengio, and Hossein Hajimirsadegh. 2024. Were RNNs All We Needed? arxiv.org.

Flynn, James R. 2009. What Is Intelligence?: Beyond the Flynn Effect. Cambridge University Press.

GitHub. 2023. Research: Quantifying GitHub Copilot's impact on code quality. October 10. https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-on-code-quality/.

Google Gemini Team. 2023. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arxiv.org.

Google Gemini Team. 2023. Gemini: A Family of Highly Capable Multimodal Models. arxiv.org.

Gu, Albert, and Tri Dao. 2023. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arxiv.org.

Gunasekar, Suriya, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, et al. 2023. Textbooks Are All You Need. arxiv.org.

Guyton de Morveau, Louis Bernard, Antoine Lavoisier, Claude-Louis Bertholet, and Antoine-François de Fourcroy. 1787. "Mémoire sur le développement des principes de la nomenclature méthodique." Méthode de nomenclature chimique 31.

Haier, Richard J. 2017. The Neuroscience of Intelligence. Cambridge University Press.

Harding, William, and Matthew Kloster. 2023. Coding on Copilot: 2023 Data Shows Downward Pressure on. GitClear.

Hendrycks, Dan. 2023. Natural Selection Favors AIs over Humans. arxiv.org.

Henighan, Tom, Shan Carter, Tristan Hume, Nelson Elhage, Robert Lasenby, Stanislav Fort, Nicholas Schiefer, and Christopher Olah. 2023. Superposition, Memorization, and Double Descent. Anthropic.

Herrnstein, Richard J., and Charles Murray. 1996. The Bell Curve: Intelligence and Class Structure in American Life. Free Press.

Ho, Anson, Tamay Besiroglu, Ege Erdil, David Owen, Robi Rahman, Zifan Carl Guo, David Atkinson, Neil Thompson, and Jaime Sevilla. 2024. Algorithmic progress in language models. arxiv.org.

Hoffman, Donald D. 2008. "Conscious Realism and the Mind-Body Problem." Mind & Matter 6 (1): 87–121.

Hopcroft, John. 2008. Introduction to Automata Theory, Languages, and Computation. Pearson India; 3rd edition.

Howe, Nikolaus, Ian McKenzie, Oskar Hollinsworth, Michał Zajac, Tom Tseng, Aaron Tucker, Pierre-Luc Bacon, and Adam Gleave. 2024. Effects of Scale on Language Model Robustness. arxiv.org.

Hubert, Kent F., Kim N. Awa, and Darya L. Zabelina. 2024. "The current state of artificial intelligence generative language models is more creative than humans on divergent thinking tasks." Scientific Reports 14 (1): 3440.

Bibliography

Hwang, Dongseong, Weiran Wang, Zhuoyuan Huo, Khe Chai Sim, and Pedro Moreno Mengibar. 2024. TransformerFAM: Feedback attention is working memory. arxiv.org.

Jiang, Albert Q., Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, et al. 2024. Mixtral of Experts. arxiv.org.

Kandel, Eric R., John D. Koester, Sarah H. Mack, and Steven A. Siegelbaum. 2021. Principles of Neural Science. McGraw Hill / Medical; 6th edition.

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. arxiv.org.

Kaufman, Alan S., and Elizabeth O. Lichtenberger. 2005. Assessing Adolescent and Adult Intelligence, 3rd Edition. Wiley.

Kejriwal, Mayank, Henrique Santos, Alice M. Mulvehill, Ke Shen, and Deborah L. McGuinness & Henry Lieberman. 2024. Can AI have common sense? Finding out will be key to achieving machine intelligence. October 7. https://www.nature.com/articles/d41586-024-03262-z.

Keles, Betul, Niall McCrae, and Annmarie Grealish. 2019. "A systematic review: the influence of social media on depression, anxiety and psychological distress in adolescents." International Journal of Adolescence and Youth 25 (1): 79–93.

Kleidon, Axel. 2010. "Life, hierarchy, and the thermodynamic machinery of planet Earth." Physics of Life Reviews 7 (4): 424-460.

Klein, Balázs, and Kristof Kovacs. 2024. "The performance of ChatGPT and Bing on a computerized adaptive test of verbal intelligence." PLoS One 19(7).

Kohlberg, Lawrence. 1984. The psychology of moral development: the nature and validity of moral stages. San Francisco: Harper & Row.

Kosinski, Michal. 2024. "Evaluating large language models in theory of mind tasks." Proceedings of the National Academy of Sciences 121 (45): e2405460121.

Krokstad, Steinar, Daniel Albert Weiss, Morten Austheim Krokstad, Vegar Rangul, Kirsti Kvaløy, Jo Magne Ingul, Ottar Bjerkeset, Jean Twenge, and Erik R Sund. 2022. "Divergent decennial trends in mental health according to age reveal poorer mental health for young people: repeated cross-sectional population-based surveys from the HUNT Study, Norway." BMJ Open 12 (5): e057654.

Kuhn, Thomas S. 1962. The Structure of Scientific Revolutions. University of Chicago Press; Fourth edition.

Lai, Wen, Mohsen Mesgar, and Alexander Fraser. 2024. LLMs Beyond English: Scaling the Multilingual Capability of LLMs with Cross-Lingual Feedback. arxiv.org.

LCM Team, Meta, Loïc Barrault, Paul-Ambroise Duquenne, Maha Elbayad, Artyom Kozhevnikov, Belen Alastruey, Pierre Andrews, et al. 2024. Large Concept Models: Language Modeling in a Sentence Representation Space. arxiv.org.

Levy, Steven, and Yann LeCun. 2023. How Not to Be Stupid About AI, With Yann LeCun. December 22. https://www.wired.com/story/artificial-intelligence-meta-yann-lecun-interview/.

Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, et al. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arxiv.org.

Li, Yuanzhi, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks Are All You Need II: phi-1.5 technical report. arxiv.org.

Liu, David, Virginie Do, Nicolas Usunier, and Maximilian Nickel. 2023. Group fairness without demographics using social networks. arxiv.org.

Bibliography

Metz, Rachel. 2024. OpenAI Scale Ranks Progress Towards 'Human-Level' Problem Solving. July 11. https://www.bloomberg.com/news/articles/2024-07-11/openai-sets-levels-to-track-progress-toward-superintelligent-ai.

Minsky, Marvin. 2007. The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind. Simon & Schuster; Reprint edition.

Moravec, Hans. 1990. Mind Children: The Future of Robot and Human Intelligence. Harvard University Press; Reprint edition.

Morris, Meredith Ringel, Jascha Sohl-dickstein, Noah Fiedel, Tris Warkentin, Allan Dafoe, Aleksandra Faust, Clement Farabet, and Shane Legg. 2024. Levels of AGI for Operationalizing Progress on the Path to AGI. arxiv.org.

Naslund, John A., Ameya Bondre, John Torous, and Kelly A. Aschbrenner. 2020. "Social Media and Mental Health: Benefits, Risks, and Opportunities for Research and Practice." Journal of Technology in Behavioral Science 5 (3): 245-257.

Noble, Denis. 2017. Dance to the Tune of Life: Biological Relativity. Cambridge University Press.

NVidia. 2014. NVIDIA® NVLink TM High-Speed Interconnect: Application Performance. NVidia.

OpenAI. 2023. GPT-4 Technical Report. arxiv.org.

OpenAI. 2024. "OpenAI o1 System Card."

Ouyang, Long, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, et al. 2022. Training language models to follow instructions with human feedback. arxiv.org.

Palmer, Jason. 2013. Entropy law linked to intelligence, say researchers. April 23. https://www.bbc.com/news/science-environment-22261742.

Pandey, Ruchika, Prabhat Singh, Raymond Wei, and Shaila Shankar. 2024. Transforming Software Development: Evaluating the Efficiency and Challenges of GitHub Copilot in Real-World Projects. arxiv.org.

Peng, Bo, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, et al. 2023. RWKV: Reinventing RNNs for the Transformer Era. arxiv.org.

Peng, Sida, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. 2023. The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. arxiv.org.

Piaget, Jean. 1932. The Moral Judgment of the Child. Routledge.

Ren, Richard, Steven Basart, Adam Khoja, Alice Gatti, Long Phan, Xuwang Yin, Mantas Mazeika, et al. 2024. Safetywashing: Do AI Safety Benchmarks Actually Measure Safety Progress? arxiv.org.

Sapolsky, Robert M. 2023. Determined: A Science of Life without Free Will. Penguin Press.

Saxe, Glenn N., Daniel Calderone, and Leah J. Morales. 2018. "Brain entropy and human intelligence: A resting-state fMRI study." PlosOne.

Schacter, Daniel L. 2021. The Seven Sins of Memory Updated Edition: How the Mind Forgets and Remembers. Mariner Books.

Ševo, Igor. 2023. Consciousness, Mathematics and Reality: A Unified Phenomenology. PhilPapers.

Ševo, Igor. 2023. Intelligence as a Measure of Consciousness. arxiv.org.

Silberschatz, Abraham, Peter B. Galvin, and Greg Gagne. 2021. Operating System Concepts. Wiley; 10th edition.

Silver, David, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, et al. 2017. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. arxiv.org.

SimpleBench Team. 2024. SimpleBench: The Text Benchmark in which Unspecialized Human Performance Exceeds that of Current Frontier Models. SimpleBench.

Bibliography

Sipser, Michael. 1996. Introduction to the Theory of Computation. Course Technology Inc; 3rd edition.

Slotnick, Scott D. 2017. Cognitive Neuroscience of Memory. Cambridge University Press.

Smil, Vaclav. 2018. Energy and Civilization: A History. The MIT Press.

Stojnić, Gala, Kanishk Gandhi, Shannon Yasuda, Brenden M. Lake, and Moira R. Dillon. 2023. "Commonsense psychology in human infants and machines." Cognition 235.

Sutton, Richard. 2019. The Bitter Lesson. March 13. http://www.incompleteideas.net/IncIdeas/BitterLesson.html.

Tanenbaum, Andrew, and Herbert Bos. 2014. Modern Operating Systems. Pearson; 4th edition.

Tononi, Giulio, Melanie Boly, Marcello Massimini, and Christof Koch. 2016. "Integrated information theory: from consciousness to its physical substrate." Nature Reviews Neuroscience 17: 450–461.

Villalobos, Pablo, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2022. Will we run out of data? Limits of LLM scaling based on human-generated data. arxiv.org.

Vogels, Emily A., and Colleen McClain. 2023. Key findings about online dating in the U.S. February 2. https://www.pewresearch.org/short-reads/2023/02/02/key-findings-about-online-dating-in-the-u-s/.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arxiv.org.

Wissner-Gross, A. D., and C. E. Freer. 2013. "Causal Entropic Forces." Physical Review Letters 110 (16).

Wittgenstein, Ludwig, C. K. Ogden, and Bertrand Russell. 1981. Tractatus Logico Philosophicus. Routledge.

Wong, Dakota, Austin Kothig, and Patrick Lam. 2022. Exploring the Verifiability of Code Generated by GitHub Copilot. arxiv.org.

Yao, Shunyu, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. arxiv.org.

Yao, Shunyu, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. Arxiv.org.

Zou, Andy, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and Transferable Adversarial Attacks on Aligned Language Models. Arxiv.org.

Zsila, Ágnes, and Marc Eric S. Reyes. n.d. "Pros & cons: impacts of social media on mental health." BMC Psychology 11 (1): 201.

Igor Ševo | HTEC

www.igorsevo.com | www.htec.com