Intelligent Agents for Information Augmentation Using Novel Approaches to Data Analysis

Rommel Martínez rommel.martinez@valmiz.com
Valmiz https://valmiz.com

October 11, 2021

One of the holy grails of Artificial General Intelligence (AGI) is true comprehension. It would hypothetically function without the use of corpora; and morphological and semantic analysis. More importantly, it would require a certain level of access consciousness (A-consciousness) [2] and awareness of causes, effects, association, relevance, immediacy, and consequences.

In this paper, we're going to tackle the challenges that prevent us from scratching the surface of AGI; the mainstream approaches to AI; the methods that will bridge us towards achieving a more general form of AI; and how we're going to ultimately find the junction points between structured and unstructured data. In this paper we present Valmiz—a collection of agent-based knowledge and discovery systems; the methods and techniques they use to approach the low-hanging fruits of AGI; and the principles and concepts which can aid in creating the implementations and structures to facilitate the discovery of early true machine comprehension, with the use of modern computing technology.

1 Introduction

The vast majority of the way artificial intelligence is being approached now is through statistical and probabilistic methods. Large data sets are being used to train systems to emulate a narrow subset of the way humans think and solve problems. These intelligent machines are only able to produce meaningful results from existing data. We have seen systems that are able to render faces of humans that do not exist in real life. We have seen systems that learned how to punch and kick. The common problem with such systems is that they rely on existing data in order to simulate learning new skills. The issue that arises from that is confirmation bias—the systems are only able to produce results based on what they already knew. We, humans, would only like to believe that they are producing something completely novel because we already conditioned ourselves to accepting them, beforehand. Truly unique composition is absent.

The way humans learn to communicate using languages and subsequently understanding them, however, is different. Let's take the case of a human child. One doesn't teach the parts of speech nor the relationship of the language components to them. They learn to communicate using a gradual learning approach—one that involves continual exploration. A child uses the constant feedback loop between them and another communicator. By having a rapid and fluid loop, a child's association with sounds becomes associative and causative with the environment that they are experiencing and perceiving. In a similar vein, if you were teaching a child how to open a door, you wouldn't open the door for him and then describe at length how the door looked when it was open. No, you would teach how to turn the doorknob so that he could open the door himself [9].

2 Background

It is easy to make a computer display adult-level performance when given tasks like solving board games, but it is impossible to make them display the abilities of a typical one-year-old when handling problems about perception and seeing the world around them from a zeroth position [7]. The main lesson of thirty-five years of AI research is that the hard problems are easy and the easy problems are hard. The mental abilities of a four-year-old that we take for granted—recognizing a face, lifting a pencil, walking across a room, answering a question—in fact solve some of the hardest engineering problems ever conceived [8].

Contemporary Natural Language Processing (NLP) systems work by using training models and existing data sources to teach a machine what are the Parts of Speech (PoS) and Universal Dependencies (UD). Such systems are able to tag an input text what are nouns, verbs, etc. because it already knows about them, beforehand. By ingesting huge corpora and comparing the results of analyzing them with another data set, these systems became good at identifying such things.

Because of the way such NLP systems work, a significant majority of them are designed to only handle the most common spoken languages—English, Arabic, Chinese, French, German, and Spanish. Corpora for these languages are not only abundant but also has a long history. Because of this, it makes it easy to create training models. The problem, however, is that text processing becomes limited to data that is available. This implies that a system trained to handle and recognize an X set of languages, will have difficulties and produce inaccurate results when tasked to handle languages outside of those sets.

Another prevalent issue with NLP systems now is whether they truly understand the text or they have merely run its input through a processor. To truly understand, in this context, means to have the comprehension skills of an average adult human. It also implies that an equivalent mental model is created based on the inputs that it has received. The problem is particularly evident with the Chinese room argument [11]. It supposes that a closed room exist with two slots on the outside—one for questions and another for answers. A questioner would slide in a piece of paper that contains Chinese text, and on the other slot comes out the answers. Inside the room lives an operator who doesn't understand Chinese, only understands English, and has a manual for written in English for matching questions to answers. The manual says that if he sees Chinese characters that match a certain shape and sequence, he would respond with the specific matching Chinese text found in the manual, using the answer slot on the room. From the questioner's standpoint, whatever is inside the room possesses the ability to both understand and speak Chinese.

Let's take the case of Sophia[1], the robot that was developed by Hanson Robotics under the guidance of Ben Goertzel. When she debuted, it was made to appear that she possessed human-level intelligence and that she'd be able to converse like a human to another human. It was also shown that she's able to convey facial expressions and body gestures, to go along with her speech. It was soon discovered that she's not any different from a marionette—human operators were necessary in order for her to operate "correctly." For whatever it is worth, she's a chatbot with a face [3].

Several morphological systems have been designed in the past decade. They approach linguistics via the textual representations of language and, that text is most often dissected into parts and how they relate to each other. Systems such as CoreNLP[2] and spaCY[3] handle linguistic interactions using morphological syntactic analysis of corpora. In addition to that, they have strong a dependence on ontological databases of what constitutes components. These systems are not able to operate inside a vacuum. They need information stored elsewhere in order to begin processing knowledge. They need seed knowledge.

Most, if not all, language systems rely on using information that has been secured beforehand—frontloading. They work exclusively using the answer model, wherein they already know the answer before the question has been asked. There is no process of inquiry. There is no curiosity. They display a certain degree of intelligence, but this is mostly due to the confirmation bias of humans, making ourselves believe that it they indeed possess cognizance, even when it is not present.

According to Noam Chomsky, humans have the predisposition to learn languages, that is, the ability to learn languages is encoded in our brains long before we are born [4, 5]. The hypotheses of Chomsky state that the reason why humans, especially children, are able to pick up language easily is that our brains have already been wired to learn it. He argues that even without the basic rules of grammar, our brains are still well adapted to learn them along the way.

In this paper, we challenge the positions of Chomsky about the innateness of learning languages. We believe that by resigning to the idea that language can only be learned innately, we lose the ability and the curiosity to understand it from its most primary underpinnings. When we commit to the idea that there's only one exclusive, golden way to learn languages, we throw away all the possibilities of effectively capturing it and properly systematizing and controlling its very nature. We believe that Chomsky's Language Acquisition Device (LAD) can be synthetically created and be installed to an empty artificial brain.

One of the key questions to raise with language learning is that can it be sped up? Normally, it would take time for a child to acquire a basic language skillset before they can communicate with the immediate people around them. Now, can a machine learn languages faster than a child? In order for AI systems to even remotely approach the A-consciousness of a two-year-old child, it must be able to communicate bidirectionally with the external world. It must be able to pose questions. It must be curious on its own. Modern AI systems can't and don't ask, to humans or to fellow machines. They can't dream. We will change that.

3 Embodiment

It is considerably more difficult to build a synthetic brain from scratch or to simulate the concept of a mind that can readily interact with the world around it—much like a four-year old child, *a priori*—than to provide a means for a learning system to interact with the world—or a subset of it—physically. Physical in this sense means being able to use sensory inputs to validate existing knowledge, capture new data, to be familiar with new inputs, and stash unknown things for later processing.

A machine now would be happy to chuck truckloads of data and assign meaning to them. The problem with this approach, however, is that the meaning does

[1] https://en.wikipedia.org/wiki/Sophia_(robot)

[2] https://stanfordnlp.github.io/CoreNLP/

[3] https://spacy.io/

not come from the machine itself but rather comes precomposed from human processing. It may be able to categorize and differentiate dogs from cats, but intrinsically, it doesn't know what they are beyond their representations as images stored on a computer system. A system based on machine learning may be able to recognize a cat in a picture, but when asked what happens when you startle a cat, it fails miserably.

The premise of embodiment is that that a machine cannot attain human-level intelligence without having some kind of body that interacts with the world. In this view, a computer sitting on a desk, or even a disembodied brain growing in a vat, could never attain the concepts necessary for general intelligence. Instead, only the right kind of machine—one that is embodied and active in the world—would have human-level intelligence in its reach [6].

With the ideas of embodiment, it is possible to construct sophisticated systems using initial embodied entities, who are going to interact with the world, like humans, but to a significantly less detailed resolution, which has the ability to transfer knowledge to disembodied systems one of its goals. In that way, embodied systems will function as both learning scouts and learning individuals. In contrast to human learning, the transfer of memes from a parent to a child takes a significantly large amount of time because of the lack of bandwidth in the brain of a child. In addition to that, the child still has to perceive the world around them, in person, to learn new things.

With that in mind, the embodied-disembodied pairing is proposed because we can take advantage of the advances in technology to transfer information unidirectionally, rapidly. Using this approach, a disembodied system may not need to interact with the world in order to process information because an embodied entity is already doing the processing of raw sensory physical inputs from the world, for the disembodied one.

4 Minimal brain

In trying to approach one of the key problems of AGI—A-consciousness, adaptability, and comprehension—it is tempting to implement all the features that allow a human to interact with other humans and with the rest of the world. Capabilities such as vision, hearing, olfaction, sense of taste, sense of touch, and mobility all contribute to enabling a human to acquire and share knowledge, test hypotheses, conduct experiments, make observations, and travel to new places. Because of these features, it makes learning very fast and natural for humans. It also forms the cornerstones of A-consciousness and reasoning. This is contrast to the handling the more difficult problems of AGI—phenomenal consciousness (P-consciousness), which deals with moving, colored forms, sounds, sensations, emotions and feelings with our bodies and responses at the center [2].

It is worth noting, however, that even if some senses are not available, a human can still mature and have sound modes of reasoning. If a man is blind at birth or becomes blind in the course of his life, it is still possible for him to practice strong reasoning, human-to-human interaction, and curiosity. If a man loses the sense of smell and hearing, he is still able to make use of the other senses to interact with the world. There are capabilities, however, that one must absolutely have in order to have a functional life, like sense of touch and mobility.

A hypothetical minimal brain would contain only the minimum processing requirements in order to process touch and execute mobility. With the sense of touch, an embodied system would be able to sense physical objects and create maps of them in its brain. With the sense of touch, an embodied system would be able to correctly qualify the properties of physical objects around him. With mobility, even if an embodied system with bipedal locomotion loses a leg, it will still be able to process inputs in its environment if it balances on one leg or move with the assistance of a tool.

Inside a virtual reality (VR) world, a disembodied system would be stopped from running if it hits a wall, not because the wall has innate qualities that prevent things from passing through it, but because of predetermined rules inside that world. An embodied system with a minimal brain would be able to explore the world and see that if it tries to walk past a wall, it is stopped. This is similar in concept to a Roomba wherein it creates a map of its environment by learning what it can pass through and what it can't.

Instead of waiting for the outstanding problems of sensory processing to be solved, a minimal brain can already be designed, whose primary attributes are having the minimal amount of sensory processors to be able to interact with the world as embodied systems. The design of a minimal brain is that it should be able to accept new ways of processing input—such as strong Computer Vision (CV)—in the future.

5 State of affairs

One of the most important components of current AI systems is data and how they're being dissected, processed, and analyzed. How data is analyzed between intelligent systems is what makes the difference. Some take the approach of pouring data into a pot, stirring it, and hoping that whatever comes out of it would make sense to a human. Others concoct fancy rules into how it must be interpreted, taking the opposite approach. The systems that we are building take inspiration from both camps but add the flexibility of making the knowledge that it has acquired to be malleable.

Currently, AI systems have training models that will try to cover all possible present and future scenarios. It does so via the use of neural networks and variations of it. Such networks are commonly observed with machine learning (ML), wherein training models are used to build a network. Usually, ML requires a lot of data to create a reasonable system to perform well. This approach is already being employed in fields from agriculture to speech recognition. ML excels at developing statistical models. However, one of the most common problems of ML is that it is unable to cope with situations that it has not been trained with. There have been numerous incidents of self-driving cars that crashed into pedestrians, trees, and overturned trucks. Black swans are ignored.

Another form of an AI system that is still in use today is Good Old Fashioned Artificial Intelligence (GOFAI). One approach of GOFAI is through the use of symbols to represent things and concepts. Trees and nodes of connections are formed to create the relationships between these symbols. In addition to connections, properties of symbols can be encapsulated inside such symbols. GOFAI excels when logic and reasoning can be readily applied to a problem domain. However, GOFAI fails when the rules that are created are not sufficient to describe a scenario. It fails when relationships between symbols cannot be determined beforehand.

Finally, a less popular approach to AI that is still in use are robots using human brain simulation. They mimic, to a certain degree, how the nervous system works. It works through the use of sensors to detect temperature, hardness, obstacles, light, and odor. These systems performed well when navigating rooms and performing factory assembly tasks. Soon after, it was realized that the intelligence that these robots possessed were fairly limited and only performed one-way tasks.

6 Data processing

Due to limitations of existing approaches to artificial intelligence, and the way we would like to handle the things where there are no elegant solutions, yet, we devised alternative methods to bridge the gaps between symbolic, sub-symbolic, robotic, and statistical learning. In order to resolve the difficulties present in these systems, it was imperative to determine whether the core concepts of each can be carried over to a new system, and whether they can be forged to work together [10].

Data can be roughly divided into two camps: structured and unstructured. It is still a subject of debate, to this day, what should be constituted as such. Most researchers would agree, however, that structured data are the ones with a uniform set of structures and can be parsed without too much ambiguities. Examples of structured data would be key-value stores, spreadsheets, and tabular data. Unstructured data, on the other hand, are the ones without a clear form, or more specifically, ones whose form cannot be easily represented in a structured manner. Examples of unstructured data are narrative text, images, and video.

The vast majority of unstructured data are still being handled through brute force, via one or more forms of neural networks. Data is still processed with human evaluators at the end, which unintentionally gives it a bias towards human inclinations—it may make sense to humans but not necessarily to other forms of life that may also exhibit intelligence. When neural networks are used to handle natural languages, the language constructs are nothing but just a mixed soup of ingredients to the system. NLU systems have no intrinsic knowledge of the information that they are processing.

With a plethora of raw data at our disposal, it becomes tempting to use these vast amounts of data to attack the language problem. The problem with this is that it's the wrong problem that is being attacked. What should we be focusing on instead is the comprehension problem. No amount of raw data is ever going to give a supposedly intelligent system intelligence without addressing the problems of understanding, first.

7 Alternative approaches

When dealing with the problems of information representation, it's imperative to determine what are the key data structures and algorithms to use. In software domains like conventional relational and key-value databases, compression, image processing, etc., it's relatively easy to pick a data structure that is already in widespread use. In those industries, the high ceilings are relatively within reach. In AI, however, it is detrimental to use data structures that are not custom-fit to handle the problems within that domain.

In trying to discover what should be the key qualities of a novel data structure that will support the kinds of capabilities that we would like to have, we have to answer the following questions:

- How is information represented?
- How is it structured?
- What kinds of data can be encapsulated?
- What kinds of operations are possible?
- What are its key features?
- What distinguishes it from other approaches?
- How can it be used?
- Are there systems that already implement it?

8 Volumes

Volumes are novel data structures groups that make it possible to perform computations, analysis, and discovery, in a way that was not easy to do before. With volumes come the concepts of *frames*, *pools*, *units*, and *cells*. Together they make up microcosms within *registries* and *universes*.

Volumes are represented as semi-contiguous connections of frames, which could either be pools or units. A frame is a container and pointer that contains navigational information in a volume. A pool is a frame that contains a value, while a unit is a frame that doesn't contain a value. A "value" in this sense means any kind of data, a pointer to another frame, or a pointer to another volume. This is the *container* property of volumes.

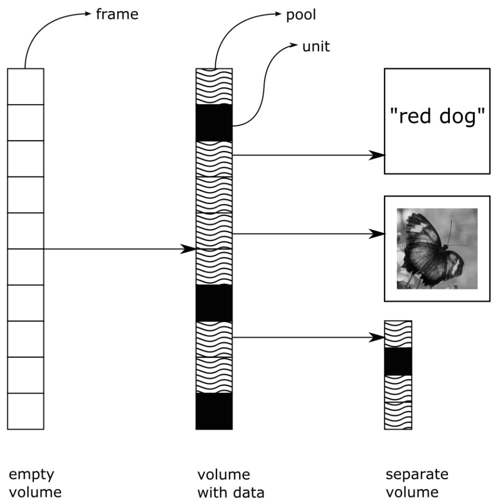

Figure 1: *Basic volume structure*

Volumes can be disassembled and reassembled in different configurations including, but not limited to: *frame burying*—the ability to temporarily make a frame inaccessible in a volume:

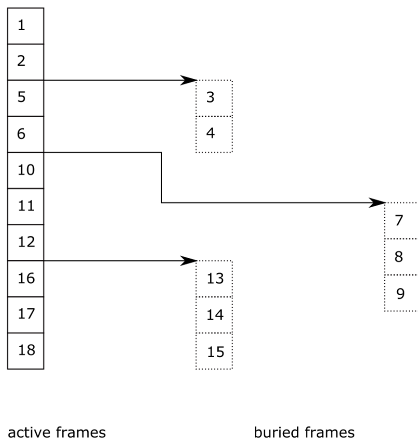

Figure 2: *Frame burying*

Frame banishing—the ability to send frames to the *void*. The void is a place where volumes and frames may still exist, however, they're not considered part of the universe while they're there. Special procedures are in place to make sure that they do not clash with the existence of volumes in the universe.

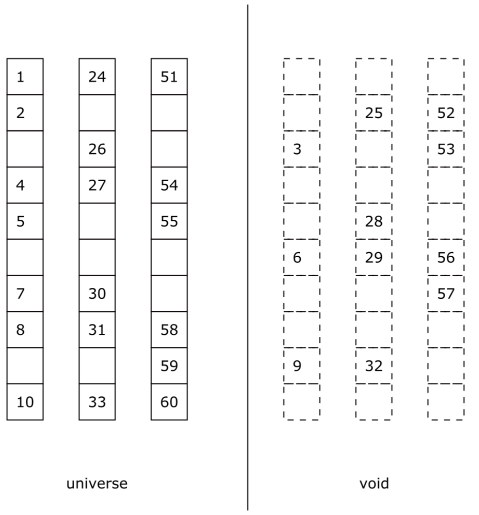

Figure 3: *Frame banishing*

Horizontal volume binding—the ability to connect and bind heterogeneous types of volumes together. This gives the ability of volumes to share properties allowing for operations like matching, searching, and lateral indexing.

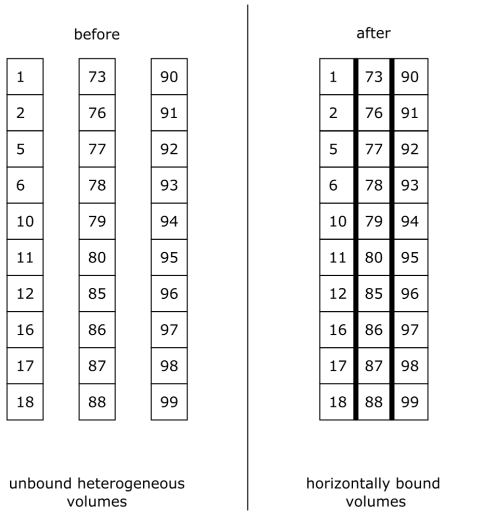

Figure 4: *Horizontal volume binding*

Vertical volume binding—the ability to bind volumes together by linking the heads and tails of different volumes. This gives the ability to extend existing properties and give more context to existing information.

Volume destructuring—the ability to decomponentize volumes into arbitrary-sized frame groups; and *volume wrapping*– the ability to create a globe of volumes, creating a monolithic volume group.

Because of the flexibility of volumes in taking arbitrary forms, we are able to make computations not possible with traditional structures. Due to the property of a volume being both a container and binder, we are able to manipulate data more dynamically and

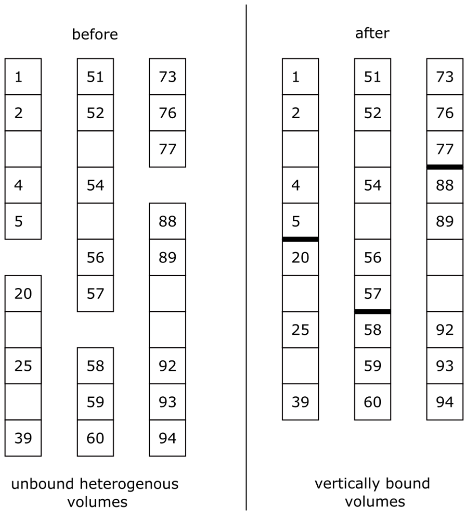

Figure 5: *Vertical volume binding*

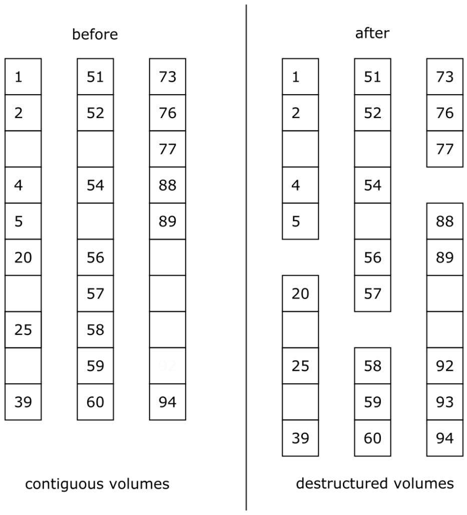

Figure 6: *Volume destructuring*

with finer grained control. Using the proper grouping of volumes, we are able to create *volume blocks*—configurations of volumes that contain specific traits and qualities. Using volume blocks, we can create a network of interrelated volume groupings.

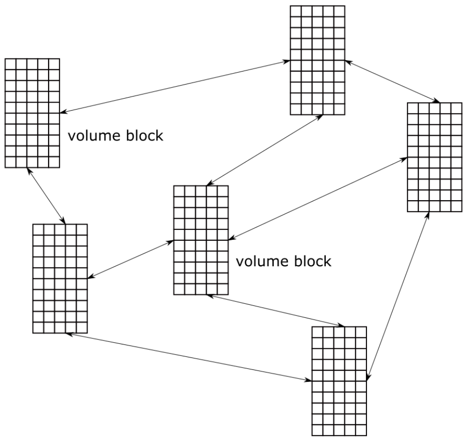

Figure 7: *Interconnected volume blocks*

9 Capsules

Capsules are storage mechanisms that allow one to manage nested encapsulated key-value information. With capsules, it is possible to create trees of relationships whilst preserving value context. When a parent or a child capsule changes property or value, the change becomes reflected in the whole subtree. The idea of capsules were inspired by earlier work on Mimix Stream Language (MSL) [1].

Each capsule can contain arbitrary data, including the value of another capsule, allowing for nesting of capsules. Information that is contained in capsules is retrieved in the order that they were defined. In Figure 8, the capsules *first* and *last* are bound to `John` and `Smith`, respectively. The capsule *alt* is bound to `Big` and the value of the capsule *first*, which is `John`. Similarly, the capsule *name* is bound to the aggregate values of the capsules *first* and *last*.

Sub-capsules on the other hand, are like capsules, but they can only live inside proper capsules. They allow data embedding, while providing a limited form of information hiding.

Constants allow bindings to a capsule that prevent new value bindings but allow for overlays on those constants to exist. Overlays provide a shadowing mechanism to constants within the environments where those constants exist. In Figure 10, a constant is created on top-level binding the value `John` to the capsule *first*. Then, an overlay is made to temporarily bind a new value to the capsule *first* inside another environment. In that environment the value of the capsule

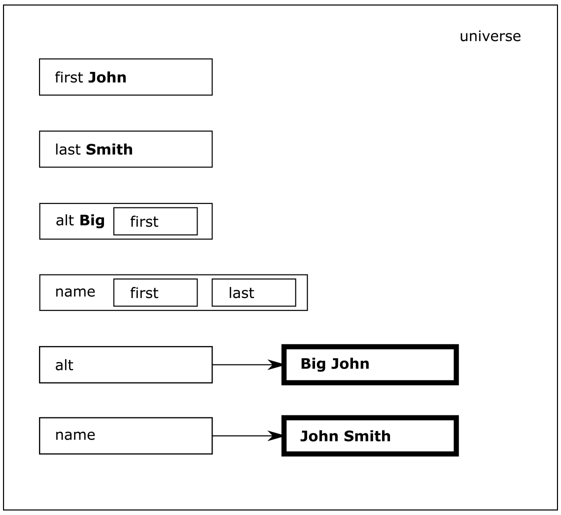

Figure 8: *Basic and extended capsules*

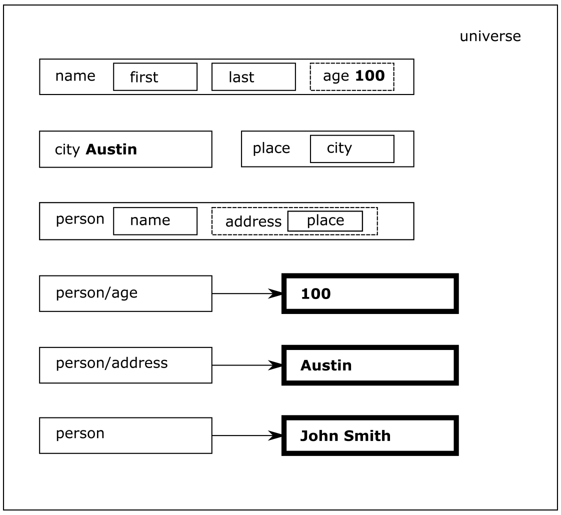

Figure 9: *Sub-capsules*

first is `Peter`. However, outside of that overlay environment, the capsule *name* is still bound to the original value of `John` when the constant was first created.

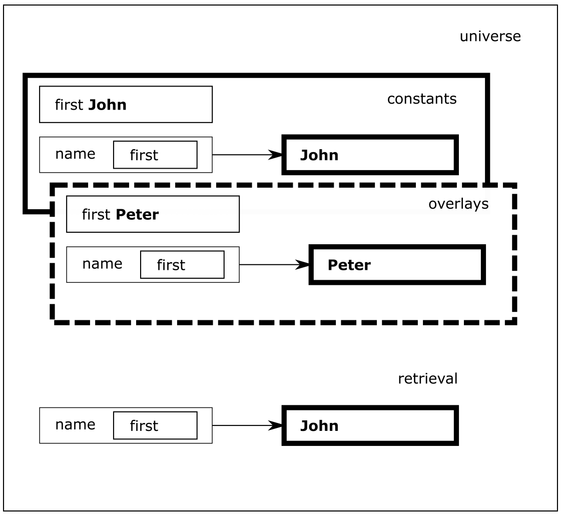

Figure 10: *Constants and overlays*

A mini-language has been designed to support the direct manipulation of the capsules–*declarations*. Declarations are user-level mechanisms to interact with the capsule system. It is a high-level language that has a similar syntax to s-expressions. A declaration can either retrieve a capsule value, set it to a new one, or overwrite the value of an existing one. At the most basic level, declarations are composed of *terms*, *sub-terms*, and *constant terms*. Terms, sub-terms, and constant terms correspond to capsules, sub-capsules, and constants in the object universe.

Terms are the basic building blocks of declarations. They can either be textual information or binary blobs. Sub-terms are terms that are inside terms. Constant terms, on other hand, are terms that do not change value inside a scope. When a new value is bound to a constant, inside another existing constant, the new, temporary value becomes the active one. When the new constant leaves the scope, the original value becomes visible again.

The declarations that correspond to Figures 8, 9, and 10 are as follows:

```
($first John)
($last Smith)
($alt Big ($first))
($name ($first) ($last))
($alt)              => "Big John"
($name)             => "John Smith"
```

Listing 1: *Basic terms*

Term names are not case-sensitive, so `($first John)` are equivalent to `($FIRST John)` and `($FiRsT John)`. Term values are implicitly quoted. Accumulation of information happens serially

```
($first John)
($last Smith)
($name ($first) ($last) :age 100)
($city Austin)
($name :address ($city))
($name :age)      => "100"
($name :address)  => "Austin"
($name)           => "John Smith"
```

Listing 2: *Basic sub-terms*

```
(%c ($first John)
    ($name ($first)))      => "John"
(~o ($first Peter)
    ($name ($first)))      => "Peter"
($name)                    => "John"
```

Listing 3: *Constant terms and overlays*

across time. All changes to a declaration are captured. This feature enables arbitrary rollbacks.

10 Implementations

To put the aforementioned ideas to practice, software that implement these data structures and algorithms was written. They were written in Common Lisp (CL)[4]. CL was chosen in order to adapt to the dynamic nature of information propagation present with volumes and capsules, to support reflective computations, and to allow seamless code updates. Due to the fact that CL is a standardized programming language[5], the source code is guaranteed to run far in the future with any standards-conforming CL system.

10.1 Veda

The canonical implementation of volumes is the Veda system. With Veda, it is possible to represent arbitrary information while making them easy to manage. With Veda, it becomes trivial to encapsulate entire worlds of information as volumes. In this way, we can approach the basic units of information as omnitraversable—one can traverse all the places in the universe in all the possible directions.

Any data group that is ingested into Veda effectively becomes a searchable database in constant time. Each component of the source data is indexed in that database. Another set of algorithms is used to make comparisons between datasets, determining similarities, differences, occurrences, ambiguities, frequency, and duplicates.

10.2 Vera

The canonical implementation of capsules is the Vera system. With Vera, it is possible to capture textual information while allowing them to be composable, dynamic, and reactive. The Vera system can be used to compose text containing authoritative information whilst allowing temporary changes. This means that the Vera system can be used as a combination of a free-form dictionary, encyclopedia, and narrative text. When applied to documents, they essentially become a living, breathing entity—the information contained there adapts to changes to adapt to the changes of contents and references.

Vera uses Veda as the backing store and serialization platform. This enables Vera to take advantage of the volume system in Veda to perform sophisticated operations not possible with traditional storage and serialization mechanisms.

10.3 Vix

To facilitate interaction with the outside world, a supervisor system codenamed Vix is being developed. Vix has several purposes. First, it acts as an interface to a human operator. It receives instructions from a user then responds back with the results of the operation. The command given to Vix can either be in textual or voice forms. Second, it acts as the primary multi-agent system that is dispatched to perform commands. When a command is received, a swarm of Vix agents is deployed to discover things and to solve problems. During this process, the members of the swarm communicate with other relaying the results of their computations. After this process, the results are collected and unified and presented to a human user or to another swarm.

Vix makes use of both Veda and Vera to achieve these things. Veda is mainly used as the primary data store and program center, while Vera is used for storing information about the things that the swarms collect.

11 Closing remarks

It is worth mentioning that the ultimate goal of these systems is not to settle the hard problems of AGI but to try to solve the parts that can be computed using contemporary computer systems. With the use of modern technology, we hope to reach a degree of consciousness that is sufficient for generating a kind of collective consciousness using multiple small agents. We like to think of these things in terms of honeybees—individually and without the connection to the hive, they behave rather simplistically. However, with a rabble, they are able to form a collective consciousness—a hive mind.

[4] https://lisp-lang.org

[5] http://www.lispworks.com/documentation/HyperSpec/Front/index.htm

Thanks to Kamil Shakirov, Chris Petersen, and Carlo Poblete for reviewing the draft of this whitepaper.

Bibliography

[1] David Bethune. “Mimix Stream Language”. In: (Feb. 2020). URL: https://web.archive.org/web/20211130084453/https://dfon51l7zffjj.cloudfront.net/uploads/company_attachment/file/48066-5ExEO7w16yWK5L3Q2T7kS9wV/MSL_Specifications_2020.pdf.

[2] Ned Block, Owen Flanagan, and Güven Güzeldere. *The Nature of Consciousness: Philosophical Debates*. 1st ed. Cambridge, MA: MIT Press, 1997.

[3] Dave Gershgorn. “Inside the mechanical brain of the world’s first robot citizen”. In: *QZ* (Nov. 2017). URL: https://web.archive.org/web/20210603051759/https://qz.com/1121547/how-smart-is-the-first-robot-citizen/.

[4] John Lyons. *Noam Chomsky*. revised. New York City, NY: Harmondsworth, Eng.; Penguin Books, 1978.

[5] James McGilvray. *Chomsky: Language, Mind, Politics*. 2nd ed. Cambridge, MA: Polity Press, 2014.

[6] Melanie Mitchell. *Artificial Intelligence: A Guide for Thinking Humans*. Reprint. United States: Picador, 2020.

[7] Hans Moravec. *Mind Children: The Future of Robot and Human Intelligence*. 1st ed. Cambridge, Massachusetts; London, England: Harvard University Press, 1988.

[8] Steven Pinker. *The Language Instinct: How the Mind Creates Language*. 1st ed. United States: William Morrow and Company, 1995.

[9] Ben Hogan, Herbert Warren Wind, Anthony Ravielli. *Ben Hogan’s Five Lessons: The Modern Fundamentals of Golf*. 1st ed. Digital Deen Publications, 2018.

[10] Herbert L. Roitblat. *Algorithms Are Not Enough: Creating General Artificial Intelligence*. 1st ed. Cambridge, Massachusetts; London, England: The MIT Press, 2020.

[11] John R. Searle. “Minds, brains, and programs”. In: *The Behavioral and Brain Sciences* 3 (Mar. 1980), pp. 415–457. DOI: doi:10.1017/S0140525X00005756.