



Linguistic and Mathematical Approaches to Deep Neural Networks Training and Artificial Intelligence: The Confluence of Paths for AI Development

Elisabetta Zuanelli

Professor Emeritus, Department of Management-Computing Sciences, School of Economy and Finance, University Rome "Tor Vergata", Italy.

Article Details

Article Type: Commentary Article

Received date: 25th November, 2023

Accepted date: 09th December, 2023

Published date: 11th December, 2023

***Corresponding Author:** Elisabetta Zuanelli, Professor Emeritus, Department of Management-Computing Sciences, School of Economy and Finance, University Rome "Tor Vergata", Italy.

Citation: Zuanelli, E., (2023). Linguistic and Mathematical Approaches to Deep Neural Networks Training and Artificial Intelligence: The Confluence of Paths for AI Development. *J Inform Techn Int*, 1(1): 104. doi: <https://doi.org/10.33790/jiti1100101>

Copyright: ©2023, This is an open-access article distributed under the terms of the [Creative Commons Attribution License 4.0](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

A Few Definitions and Conceptual Specifications

AI can be briefly defined as the emulation of the functions/activities of human brain/mind by the machine. This statement has generated the basic assumptions for the development of algorithms in AI and specifically, among others, for deep neural networks training. However, a semantic flaw lies in the definition of AI in terms of neural networks and specifically in the distinction between brain and mind. In fact, brain is the neuro-physiological support of mind. Therefore, in order to investigate human functions we must refer to the notion of mind. The notion of mind in cognitive psychology and cognitive linguistics/semantics refers to the human faculty of knowledge acquisition, memorization and use of information for all kinds of human activities whether physical or intellectual. Signals conveyed to the mind utilize different channels/sensors, namely what we call human senses that input noises, sounds, images, chemical stimuli, etc. to the central semantic memory. In this activity the question is: how is the meaning of single specific different inputs analysed, interpreted, stored and related to other inputs by the human mind. A fantastic question: it's what we call the meaning of meaning. A basic problem, then, concerns the meaning/value assigned to single elements/entities in mind and the meaning of their interrelation. This process is conferred to the splitting and definition of human experiences by means of mental representations translated into words. In linguistics, this faculty is what we call the semantic omnipotence of language and its metalinguistic capability to represent conceptually and linguistically all human experiences and its reflexive capability to explain itself: words to explain words. The semantic omnipotence of human language is defined as the potential capability of the human mind to create and use indefinite systems of signs/languages, whether natural or artificial ones, such as the languages of mathematics, physics or computer science, endowed with symbols, syntax, semantics, etc. Therefore the preliminary question on mathematics is: could mathematics exist without a preliminary conceptual linguistic representation of the physical/mathematical concepts to be coded? Now, let us turn to the specific application of mathematical concepts/functions to the computer science.

As we know, ever since the birth of computer science, algorithms have been used to emulate human activities by the machine in an

automatic way. An algorithm is not a program or code. It is the preliminary statement and definition of a procedure for solving a problem represented as a simple step-by-step description. In the context of computer science, then, an algorithm must be turned into a mathematical process for solving the problem using a finite number of steps. Algorithms use sets of initial data or input, process it through a series of logical steps or rules, and produce the output (i.e., decision, result, etc.). Code is the implementation of the algorithm into a specific programming language (like C++ or Python), while a program is an implementation of code that instructs a computer on how to execute an algorithm and perform tasks such as navigation systems, search engines, robotic performances, face/visual recognition, etc.

Then two other questions. Can algorithms allow computers to learn on their own? The answer is no. Can computers improve their performance on a specific task? The answer is yes provided they use correct data. Many people assume that machine learning algorithms use data to identify patterns and make predictions, etc., in a magic automatic way: algorithms need architectures and proper data. Therefore the final question is: what data sets and what the rules.

According to Gitta Kutyniok [1], a comprehensive theoretical mathematical foundation in AI is completely lacking at present. In AI, for instance, in the case of deep neural networks, "the search results is a time consuming work for a suitable network architecture, a highly delicate trial-and-error-based (training) process, and missing error bounds for the performance of the trained neural networks"[2]. Good news is that in the age of data, vast amounts of training data are easily available. However, the big step forward of mathematics in AI neural networks was to have solved the problem of dimensionality of data. Hundreds of layers of data, quality of data input, and the recent power of processing of modern computers are basic elements for AI. Yet, the problem lies in the quality of data inputs and the related neural networks architecture. A simple question: how is a data set defined, namely what is the conceptual value assigned to single data in a set and what the semantic relations of these and other data that are conveyed to the machine?

All applications of AI face the problem of data acquisition and

integration, often into a multimodal analysis (for example, the integrated interpretation of data by a drone on a military field where visual, phonic, phonetic, temporal, chemical information, etc. has to be recorded, merged and analysed).

How can this be done?

The development of a different approach [3] that does not exclude but rather includes the mathematical approach comes from ontologies on linguistic, semantic and cognitive bases. As stated by MITRE, NIST, ENISA on the need of an ontology in the cybersecurity domain, knowledge representation of data requires the analysis of specific domains, the specification of language entities as defined by means of controlled vocabularies and the definition of logical semantic relations such as properties, attributes and operational functions, translated into a machine readable format.

Some doubt that nominal data/labels can be organized to be machine interpretable, mathematically operationalized. They can, provided they are specified and located into an AI data architecture through specific metadata.

This linguistic semantic approach solves the lack of data architecture in deep neural networks based only on a mathematical construct and the lack of motivated specific semantic relations of data sets to be dealt with by the machine. Ontologies need:

1. An upper level, a mid level and a domain specification
2. The identification of conceptual linguistic entities that derives from knowledge representation of the domain
3. A controlled semantic vocabulary
4. And the definition of classes/categories related to entities
5. Both in a taxonomic and a transversal order
6. According to attributes, properties and operations
7. Translated by means of a metadata language into a machine readable format.

The conclusion is a plea for the need of a strong integration of the linguistic/semantic and mathematical approaches for the optimal development of machine learning applications and AI future evolutions.

That's what we've done in our cybersecurity domain platform prototype, Pragmemma POC, now in an experimental testing phase.

Ontologies and taxonomies: major flaws

In our comparative analysis of present and past ontologies and taxonomies of cybersecurity I've highlighted these major flaws:

- The lack of an upper-level ontology definition of entities and relations
- The intuitive listings of cybersecurity entities and their logical semantic relations
- The lack of a formal motivation and structuring of taxonomies and ontologies
- The quality of controlled vocabularies for entities definitions (if any)
- The lack of IoCs/IoAs data integration, correlation and classifications for attacks/incidents.

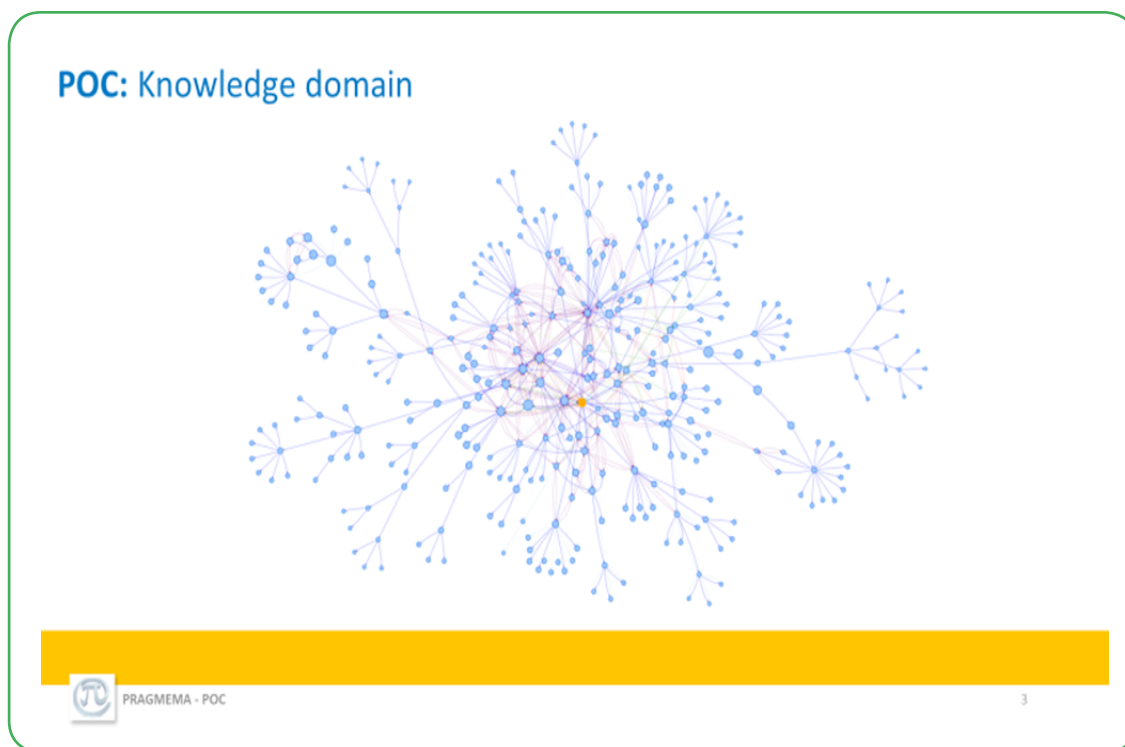
Our Pragmemma/POC prototype as presented in the NATO Conference 2022 is structured into three levels corresponding to three ontological/technological objects:

- The cybersecurityknowledge
- The cybersecurity domain ontology
- The pragmatic ontology of cybersecurity services (plus three subdomains: financial, automotive and shipping).

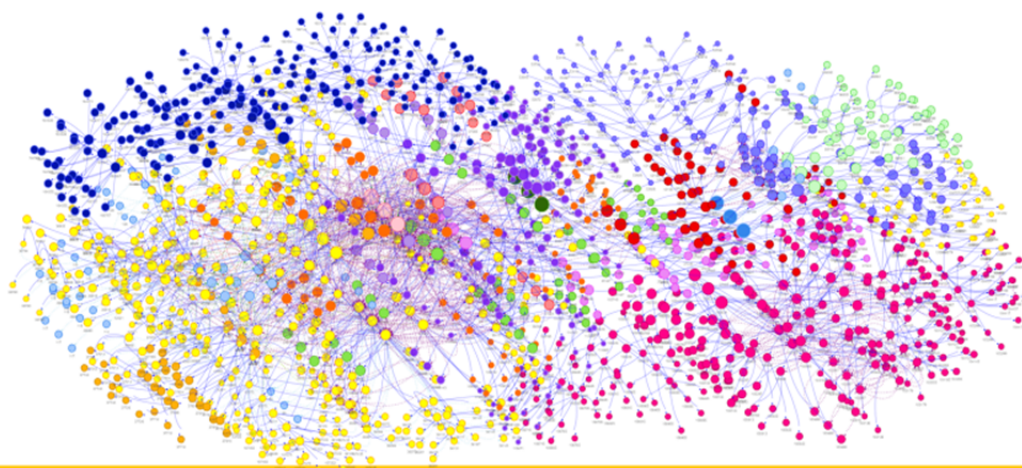
The structure can be integrated with corporate cybersecurity data in monitoring devices (IDS, IPS, firewall, antimalware, antivirus, antispam, honeypot, penetration tests, etc.) already installed or to be installed, constituting a unique level of automation and data analytics.

Here is the neural networks data base screenshots related to basic items.

The network graphs representation of the cybersecurity ontology and taxonomic classes, entities (586) and relations



POC: Cybersecurity domain ontology



PRAGMEMA - POC

4

POC: Typical incident



PRAGMEMA - POC

5

The present applications can be customized and allow for the integration and interpretation of events/incidents IOCs within the overall ontology referring to clusters of data filtering, for preventive and predictive analysis. The approach is the merging solution as prospected.

References

1. G. Kutyniok, (2022). 'The Mathematics of Artificial Intelligence', arXiv:2203.08890v1 [cs.LG] 16.
2. Id. p. 2
3. E. Zuanelli, (2022). 'Cybersecurity ontology and defense solutions: the POC platform', ScienceDirect, *Procedia Computer Science*, 205. 300–309