EasyChair Preprint

№ 7921

Hierarchical Temporal DNN and Associative Knowledge Representation

Shimon Komarovsky

August 1, 2022

Hierarchical temporal DNN and Associative knowledge representation

Shimon Komarovsky[1][0000−0002−9036−0282]

Technion - Israel Institute of Technology, cm5099@yahoo.com

Abstract. This paper proposes two models. The first one is designed bottom-up, i.e., mostly based on DL and Jeff Hawkins' temporal principle. The second one tackles some aspects of intelligence, specifically concerning the thinking process. It is designed top-down, i.e., mainly based on cognition and communication.
Additionally, this paper not only exhibits top-down verse bottom-up approaches, but also presents the two edges of evolution: the DL model considers the beginning state of learning, while the knowledge representation model considers the saturated/mature/final state of learning.

Keywords: Hierarchical · Temporal · Deep · Associative representation.

1 INTRODUCTION

An AGI design should handle a large variety of scenarios and have many vital features. Features such as: flexible, fluid, adaptive, and evolving.

We first propose a DL Model (DLM) originated mainly from the neural model in [12]. Then, we propose a model for the important components of an AGI agent: thinking and memory. It models the representation of elements in a memory, and describes how the thinking process accesses them and manipulates them for different tasks. It also encourages flexibility and adaptivity.

It is evident in neuroscience and DL that knowledge has a hierarchical structure, though there is a controversy about which type is it. In DL and [12] it is a hierarchy of features, while in [13] it is about the compositionality of objects. In this paper, our DLM is mainly established on temporal hierarchy. Whereas our knowledge representation model is based upon associative hierarchy, designated for efficient memory access.

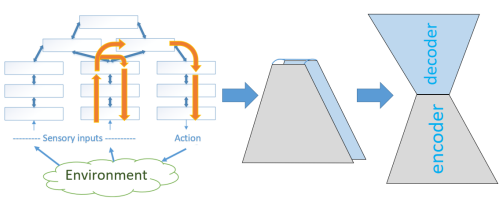
Finally, both of our presented models are based on the System 1 and 2 principle [6], on a neuro-symbolic combination by converting raw features into operational concepts, and on the stimulus-response principle, since we believe that one of AGI's characteristics is that knowledge is operational. In other words, elements that are learned are either objects or their attributes or actions which act upon them. This notion is presented in many papers on associative memory or associative NNs, where an association is a response to a stimulus, which can be either other stimuli [28] or a behavioral response (action) [15].

Please note the DLM, and especially the AKREM, are preliminary ideas. Also, the DLM is constructed from common and well defined components, and cited papers are given as suggestions to implement some of these components.

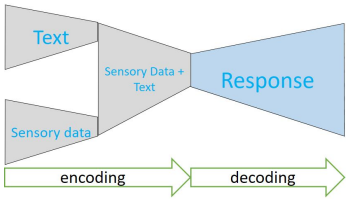
2 The proposed DLM

Our DLM is inspired by the neural models and DLMs such as caption generation [35] and Visual-Question-Answering [7]. As shown in Fig. 1(a), the idea is to unwrap the percept-predict structure from the neural model [12] on the left, into a discriminative-generative or an encoder-decoder structure on the right.

The proposed DLM is illustrated in Fig. 1(b). In this structure the data coming from text and sensors is encoded. The text includes both information and instructions. Finally, this data is encoded into some extracted features representing the whole situation, including what the model is requested to do, and then up-sample it to the actuators (the decoding process).



(a) Percept-predict structure turns into a discriminative-generative structure.



(b) Sensory data and text in encoder-decoder structure.

Fig. 1. Sensory data, text and response in the proposed DLM.

Our DLM purpose is to be able to plan and respond to inputs according to symbolic representation, which is derived from unstructured sensory data.

It is aimed to accomplish this by gradual learning in the following phases. First, it learns to fuse multi-modal inputs, to establish basic semantic concepts. Next, its objective is to learn two types of data: objects and their inter-relations. Hence, it starts by learning basic elements/objects, and then continue with composite objects and relations.

After this proper symbolic comprehension, it turns to learn how to respond correctly within the common supervised learning approach. Only it does so within several temporal resolutions: from fast to slow perception and response.

In the next section, we elaborate on these phases.

The gradual temporal representation and learning, implemented in the encoder and the decoder of our DLM, is based on the hierarchical temporal principle proposed in Jeff Hawkins' first book [12].

The proposed DLM, as any DLM, is not sufficient to serve as an AGI. Some issues are addressed in "Issues with the proposed DLM" in Appendix 4.

2.1 Proposed DLM function

A more detailed implementation of the proposed DLM is discussed.

The DLM has a hierarchical temporal structure, and it is mainly based on two ideas: the joint learning of multi-modal input, and the learning of intermediate tasks [8, 10, 14]. The latter is used to implement scene understanding within different time scales (short, mid, and long terms). For more details about these and other aspects, see in Appendix 4.

The first idea is about extracting features separately from sensors and text, then learning them together via joint embedding space [29]. Thus, the assumption here is that these inputs are complementary. Since if they are trained together, then if one of them is missing, it is sufficient for recognition as if the second one was there too. These fused features represent spatio-temporal information for the short-term temporal resolution. In the next phase of learning, these joint features are extracted further into longer time scales, by freezing first the short-term RNN layers and activating mid-term layers only. The same goes for the long-range layers afterward.

The second idea is generally about hierarchical learning of tasks [3, 11, 24], whereby several layers of tasks are learned instead of the usual single output layer of tasks. In temporal hierarchical learning, the current layer of tasks is learned first, then later more complex tasks are learned in a new layer, based on the previous tasks.

In our DLM, it is realized by intermediate tasks via RNNs. Using the first idea, the features are extracted in different time resolutions. These features are the hidden and the output layers in RNN. However, to include intermediate tasks for different time resolutions, the encoder-decoder structure of RNN is used, as in translation tasks. In other words, the intermediate tasks are connected to the context signal(s) of the RNN, not to its hidden/output signal(s). A decoder is attached to the context or to the encoder layer in the RNN. Thus, the intermediate tasks are the outputs of each of these decoders. See more in [24], and in "Hierarchical Learning" in Appendix 4.

In conclusion, there are two ways to implement hierarchical temporal learning. Either via the first idea, thus to learn multi-modal data in joint embedding space, at different time scales. Or, via the second idea, where features are extracted hierarchically temporarily (via RNN output/hidden layers), and intermediate tasks are inserted into the temporal structure. Tasks assisting in forming correct and more appropriate (guided) features, as in [3, 24, 27, 32]. Thus, after the recognition of spatio-temporal objects in the features extracted from the two inputs, their relationships should be recognized too. Hence, the intermediate tasks derive these relationships between objects. Some papers [20, 36] focus on pairwise interactions between perceived objects in an image, e.g. via a 2D graph matrix, whereas [21] models high-order interactions between arbitrary subgroups of objects.

The full sketch of the proposed DLM is shown in Fig. 2. It is seen that the decoder is also hierarchically-temporarily constructed, as a mirror image of the perceptual encoder, with skip connections, whose function may be: copy, normalization, or addition.

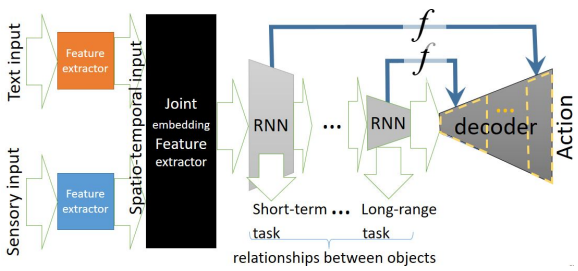


Fig. 2. Hierarchy-temporal DLM.

3 Associative knowledge representation model (AKREM)

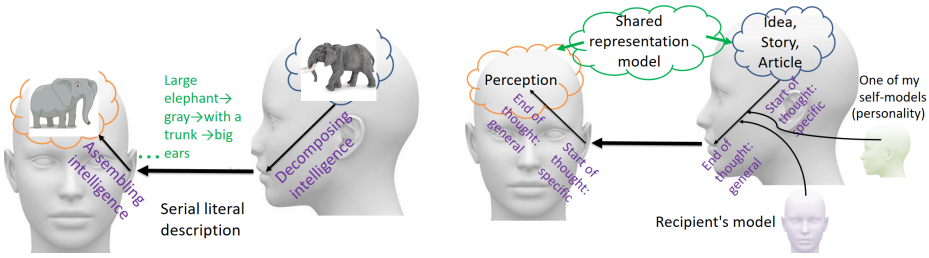
In this section, a model of AGI's knowledge representation is described. As explained in 3.3, it can utilize our general DLM, 2, as its base memory. This model tries to encapsulate a few cognitive important elements: short-term memory (STM), long-term memory (LTM), working memory (WM), and thinking. As mentioned in the abstract, it is designed in a top-down fashion. Specifically, it originates from our communication model.

3.1 Communication

Principles Our fundamental assumption about human-human communication is that each person is a "black box". Thus, we do not have access to the actual inner interpretation and representation of persons' knowledge. In other words, we communicate externally, via objective tools (the language), but we have hidden subjective perspectives or world models, constructed during a lifetime via different circumstances and experiences. This assumption is illustrated in Fig. 3(a), where the inner representation of the same message varies among people.

Next, our communication model consists of several principles. (i) The sending process is about converting an abstract message, such as a story or technical procedure, into a sequence of words. Hence, this process is generative. It is about decomposing a high-level idea into low-level concepts. Exactly opposite is the receiving process. In it, the recipient tries to assemble the idea from the low-level concepts, hence it is a discriminative process. These processes are visualized in Fig. 3(a). (ii) These couple of processes can be viewed also temporarily. The sender's thought is materialized fully when he begins his sentence(s). But to fully capture his message, the recipient has to wait till the end of the message. Hence, the end of the thought is the beginning of the message, while its start is the ending of the message. (iii) Additionally, it is about context. Due to the "black-box" assumption, to be maximally understood, the sender must start in the most general context, or common ground, to fit the message to a wide range of different recipients, with a different states of mind. And then gradually lead the recipient to his specific message. Such a chronological process would be optimal for delivering the message as accurately as possible. (iv) Finally, to

make the message clearer, both communicators should hold the models of all the relevant participants in the conversation (the recipient, the sender, their shared common knowledge, and their self-models). For principles (ii)-(iv) see Fig. 3(b).



(a) Perception and communication (b) Models in communication

Fig. 3. Communication basics

Models in communication More generally, principle (iv) reveals that human-AGI communication requires something more than merely a set of models. It requires that the AGI itself hold human-like cognitive properties and capabilities, so that humans and AGI agents would be synchronized during communication and understand each other. Hence, the AGI should have characteristics such as episodic memory, continual learning, abstraction, and generalization.

Furthermore, a more broad interpretation of principle (iv), suggest that humans are actually modeling everything. Although, we model each thing differently - depending on our interaction with it. It applies to both different people (different interactions) and different groups of people. Similarly, it applies to each object/animal or their groups. Interaction with human(s) is unique because it creates a model by conversational interaction. This idea is illustrated in Fig. 4. We probably have also self-modeling, i.e. expectations from us, in the opposite direction of the interaction. In other words, how a person should behave in different groups, with different people, and with different animals and objects. Moreover, we can model ourselves, while viewing ourselves externally (as if we are another person), to learn and perhaps change our behavior.

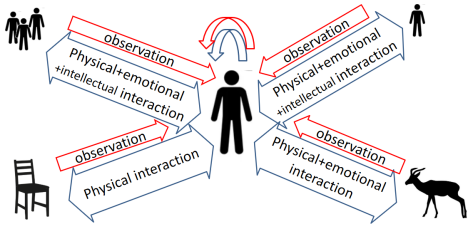


Fig. 4. Human create models from interaction.

Additionally, we perform a passive interaction, i.e. a simple observation. For example, infants mimicking when observing other humans (e.g. parents/siblings).

All the above describes the theory of embodiment, expressed by the boundaries an agent creates between different entities and between them and itself.

3.2 Detailed AKREM

Based on the communication principles above, AKREM is derived, expressing how information is represented in a memory, and to serve as a basis for cognition.

Function Our AKREM is mainly originated from two aspects: (i) the phenomenon of random bouncing from one thought to another; and (ii) the communicative hypothesis of converting an idea to low-level concepts and vice versa. This model shows how information is represented. In the decoding of a message, it is represented via the dynamic construction of hierarchical structures, similarly to constructing syntactic trees of sentences in Natural Language Processing (NLP). While in the encoding of a message, it is about descending a given hierarchy, according to a chronological order, gathering lowest-level facts and thus producing a sequence. A video demonstrating how a specific story is generating an associative hierarchy, is in: "AKREM decoding" in [16].

It can be seen in the video, that when a new unrelated piece of knowledge enters the input, the previous pieces are grouped in form of association(s). It is a bit similar to the dynamic event detection [17], where a sequence is discriminated into a set of events. As here, the task is accomplished by recognizing similarities and dissimilarities in a sequence. Only the difference is, that there is only event discrimination, while here it is about constructing a plot out of the recognized events. Moreover, the DNN stores any new (frequent enough) composite event, which results in combinatorial explosion issue, while here it does not store any combination of events as a new event. In other words, unlike dynamic event detection, which has to store and define each new combination of events, here the knowledge storage is separated into two types of memory: concepts/procedural memories to store basic events, and episodic memory, to store any new encountered combination of basic events, which is constructed dynamically.

Hence, AKREM can be considered as an upgraded model of the dynamic event detection model. The next paper extends this associative model even further, into a model of models.

Next, we formalize this model as a general structure of some plot/message. We can imagine first details about a scene are triggered one by one, and placed in level 0 of the newly generated hierarchy. Next, another scene is introduced. Each scene is represented by combining all its details in level 1. At the end of chapter 1, a few scenes were gathered. After finishing chapter 2, both chapters are connected to be in level 2. And it can go on and on. See Fig 5(b).

In order to both generate a hierarchy from a sequence or vice versa, some kind of order has to be stored, e.g. chronological/causal, in all the levels of the plot,

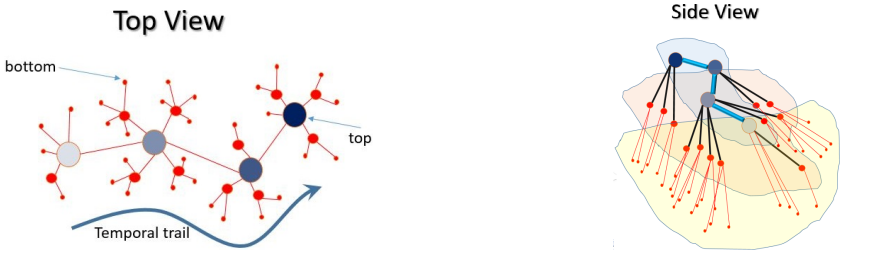
see for example the temporal trail in Fig 5(a). But the direction in connections can be extended further. It can represent different types of connections, e.g. between the levels and between the hierarchies; abstraction/generalization; various associative connections, e.g.: comparison, analogy, causality, and correlation.

It is seen that the lowest level (0) is the most general with the most objective context, since the low-level concepts have so many associations, that they lose almost entirely their specificity. However, as one goes higher in the levels, the more specific the context becomes, since it is constructed underneath a more specific structure. Hence, the highest levels hold the essence of all levels below. Thus, they possess the most accurate message.

The meaning of low-level concepts having the most associations is that they are connected to a huge amount of such hierarchies in the memory, gathered so far. The higher one goes in a hierarchy, the fewer associations it has with other hierarchies, until one reaches the levels separating this hierarchy from the rest.

Note that how the grouping occurs was not specified. For now, the grouping can be considered as summarizing or finding the essence of distinct items, but the grouping can also be treated as finding some common meaning or a purpose. See for example in the video, that for every grouping one can ask the question "why" regarding the meaning of the items in the group, whose answer is representing the grouping.

Thus we presume that our thinking is purposeful. We assume that in active/generative mode we have a purpose and we construct a hierarchy keeping in mind the purpose the whole time (perhaps in a top-down fashion), while in passive/receiving mode we construct the sender message from details, i.e. bottom-up, reconstructing its purpose.



(a) Top view of the trajectory (b) Side view of the trajectory

Fig. 5. Associative thinking via associative trajectories

At other times, we can have a no-purpose thinking. It can be viewed as a wandering between existing hierarchies, and randomly jumping from one to another, at random levels within them. This is the first aspect mentioned above.

Characteristics Associative thinking occurs all the time in our opinion.

For example, daily, where the hierarchy is constructed like a long narrative, with some experience at the top of the narrative's trajectory, made out of all separate events that occurred during this day. But it can also be attached to a previous hierarchy of the previous day, and even the previous week/month/year.

We use associative thinking in most of our cognitive tasks: in generating/perceiving a story/event/message, which is some (non-)linear plot of details; and in planning/simulating/contemplating/problem solving, which is also a series of possible actions and outcomes.

This AKREM is like a holographic memory, where the triggered neurons are shown in Fig 5(b) on the yellow surface at the bottom. They belong to the DLM presented in Fig. 1. Hence, this holographic memory is orthogonal to this DNN. In other words, we can consider triggered neurons in this DNN, producing this hierarchical dynamic structure.

We propose that the perception operation in AKREM would be similar to the one in [6]. In it, perception occurs via system 1, a multi-agent system, where agents compete parallelly with each other to decide which pattern is perceived correctly from the senses, and hence also decide which response is suitable for it. A similar idea is presented in [13], where this competition is via triggering all relevant neurons, and then filtering out all irrelevant ones as more clues are coming from the senses. Irrelevant ones predict worse than others, hence we are left eventually with the correct pattern. The process above describes recalling, hence if no pattern is recognized, a new hierarchy/memory is generated.

Both in [12] and AKREM this perception idea is expressed by ascending multiple triggered memorized hierarchies, and then descending for prediction or verification. Thus, filtering all the non-relevant memories. When partial, corrupted, or unorganized information is encountered, it can be validated not only by descending, but also by moving in all the different directions in the hierarchies. For example, in recalling a story from a scene, the agent has the freedom to move back and forth temporarily in the hierarchies.

Associative thinking/approach is much more effective than context alone, since context might consist of many details, while associations can reduce the detail level and emphasize the abstract structure of the details. Additionally, this allows for minimal communication and minimal resources in cognitive processes, enabling very few items in the WM, e.g. 7 ± 2 items.

It is important to note that AKREM is a data representation model, not yet developed into a fully working NN model. Emerging hierarchies in the WM can be implemented e.g. by some non-parametric method, such as via decision trees, since their structure is dynamic. Moreover, the number of visitations of each node and connection can be stored in these hierarchies, to distinguish this way STM from LTM.

Additionally, AKREM is a mature model, i.e., it is in the state of adulthood, which is the state reached after there has been some learning stabilization. Hence, this model also lacks the evolution of memory till its mature state. Thus, it is missing all the primary learning and adaptation. It could be fulfilled, for example, via self-supervising learning of predicting the missing/next sensory inputs.

Finally, this model has many implications, similarities with other techniques, examples, and other considerations, which should be deeply discussed in a broader paper. Additional notes (e.g. limitations and contribution): in Appendix 5.

3.3 Memories in AKREM

Besides having our associative hierarchical structures, as elements in some memory, we also should address the memory structure itself.

As in humans, systems 0,1 and 2 [25] should be realized here too. Systems 0 and 1 are expressed when the most frequent memory is used, in cases when automatic or no-thinking tasks are performed. Whereas system 2 is expressed by thinking, such as in problem solving, and it activates LTM and WM. A partial AGI model, consisting of AKREM and some DLM as its basis would also enable cases where the system is fully utilized, i.e. simultaneously thinking and performing automatic tasks.

We can assume that simple sensory perception is using base memory, similar to system 0 automatic system (no thinking), see Fig. 10. Then it provokes LTM concepts or events, "uploading" them to the WM (or STM), see Fig. 6(a). During a sleeping period, the system somehow decides what to consolidate into LTM and what not, due to unimportance or similar memories that already exist there. LTM and WM do not have direct contact with the sensors and executions, perhaps since this is abstract thinking, in which the thinking, depending on some externally-driven task, is moving in purposeful trajectories/hierarchies, mostly regardless of the inputs.

We assume that humans have permanent associative wandering in LTM, producing some final or intermediate results that are updated in WM. Differently, the wandering in AGI must have some purpose. Hence there are some external instructions inserted in this process, guiding it. See Fig. 6(a).

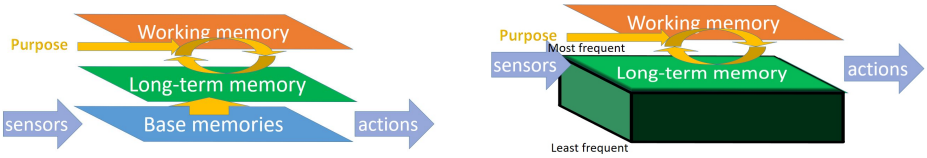
We believe that humans solve any situation/problem this way, i.e. by jumping associatively from element to element with some guiding will, searching for something, meanwhile gathering some intermediate insights, to eventually resolve with some response (good/no/bad solution).

Alternatively, we can regard the base memories, to be simply a part of the LTM. Hence, they represent the most frequent (nearly automatic) part of it. Thus, the least frequently used memory is at the bottom, while the most used memory is at a higher level, while WM serves as the currently used memory, and is located on top of this LTM unit. See Fig. 6(b).

Finally, an additional aspect of generalization is addressed in Appendix 6.

References

1. Ahmad, W.U., Chang, K.W., Wang, H.: Context attentive document ranking and query suggestion. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 385–394 (2019)



(a) Discrete memories interaction. (b) Continuous memories interaction.

Fig. 6. Memories in the associative thinking model.

2. Arel, I.: Deep reinforcement learning as foundation for artificial general intelligence. In: Theoretical Foundations of Artificial General Intelligence, pp. 89–102. Springer (2012)
3. Cerri, R., Barros, R.C., De Carvalho, A.C.: Hierarchical multi-label classification using local neural networks. Journal of Computer and System Sciences **80**(1), 39–56 (2014)
4. Cheng, H., Lian, D., Deng, B., Gao, S., Tan, T., Geng, Y.: Local to global learning: Gradually adding classes for training deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4748–4756 (2019)
5. Chomsky, N.: Aspects of the Theory of Syntax, vol. 11. MIT press
6. Daniel, K.: Thinking, fast and slow (2017)
7. Desta, M.T., Chen, L., Kornuta, T.: Object-based reasoning in vqa. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1814–1823. IEEE (2018)
8. Diao, X., Li, X., Huang, C.: Multi-term attention networks for skeleton-based action recognition **10**(15), 5326
9. Dong, J., Li, X., Snoek, C.G.: Predicting visual features from text for image and video caption retrieval **20**(12), 3377–3388
10. Ge, L., Li, S., Wang, Y., Chang, F., Wu, K.: Global spatial-temporal graph convolutional network for urban traffic speed prediction. Applied Sciences **10**(4), 1509 (2020)
11. Gu, J., Zhao, H., Lin, Z., Li, S., Cai, J., Ling, M.: Scene graph generation with external knowledge and image reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1969–1978 (2019)
12. Hawkins, J., Blakeslee, S.: On intelligence: How a new understanding of the brain will lead to the creation of truly intelligent machines. Macmillan (2007)
13. Hawkins, J., Lewis, M., Klukas, M., Purdy, S., Ahmad, S.: A framework for intelligence and cortical function based on grid cells in the neocortex. Frontiers in neural circuits **12**, 121 (2019)
14. Hwang, K., Sung, W.: Character-level language modeling with hierarchical recurrent neural networks. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5720–5724. IEEE
15. Keysermann, M.U., Vargas, P.A.: Towards autonomous robots via an incremental clustering and associative learning architecture. Cognitive Computation **7**(4), 414–433 (2015)
16. Komarovsky, S.: Playlist of agi'22 related videos, `https://www.youtube.com/playlist?list=PLvii8t7-Yebi6J25SyKbW5okEmZLME-fh`
17. Komarovsky, S.: Dynamic and evolving neural network as a basis for agi. EasyChair Preprint no. 7922 (EasyChair, 2022)

18. Leslie Smit, Pei Wang, C.H.B.R.: Can deep neural networks solve the problems of artificial general intelligence?, http://agi-conf.org/2016/workshops/
19. Li, A., Lu, Z., Guan, J., Xiang, T., Wang, L., Wen, J.R.: Transferrable feature and projection learning with class hierarchy for zero-shot learning pp. 1–18
20. Li, Y., Ouyang, W., Zhou, B., Wang, K., Wang, X.: Scene graph generation from objects, phrases and region captions. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1261–1270 (2017)
21. Ma, C.Y., Kadav, A., Melvin, I., Kira, Z., AlRegib, G., Peter Graf, H.: Attend and interact: Higher-order object interactions for video understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6790–6800 (2018)
22. Mao, J., Gan, C., Kohli, P., Tenenbaum, J.B., Wu, J.: The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. arXiv preprint arXiv:1904.12584 (2019)
23. Matsuo, Y., LeCun, Y., Sahani, M., Precup, D., Silver, D., Sugiyama, M., Uchibe, E., Morimoto, J.: Deep learning, reinforcement learning, and world models. Neural Networks (2022)
24. Nguyen, D.K., Okatani, T.: Multi-task learning of hierarchical vision-language representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10492–10501 (2019)
25. Prokopchuk, Y., Nosov, P., Zinchenko, S., Popovych, I.: New approach to modeling deep intuition. In: Materials of the 13th Scientific and Practical Conference «Modern Information and Innovative Technologies in Transport (MINTT-2021)». Kherson, Ukraine: XSMA. pp. 37–40
26. Riegel, R., Gray, A., Luus, F., Khan, N., Makondo, N., Akhalwaya, I.Y., Qian, H., Fagin, R., Barahona, F., Sharma, U., et al.: Logical neural networks. arXiv preprint arXiv:2006.13155 (2020)
27. Sanh, V., Wolf, T., Ruder, S.: A hierarchical multi-task approach for learning embeddings from semantic tasks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 6949–6956 (2019)
28. Shen, F., Ouyang, Q., Kasai, W., Hasegawa, O.: A general associative memory based on self-organizing incremental neural network. Neurocomputing **104**, 57–71 (2013)
29. Socher, R., Karpathy, A., Le, Q.V., Manning, C.D., Ng, A.Y.: Grounded compositional semantics for finding and describing images with sentences. Transactions of the Association for Computational Linguistics **2**, 207–218 (2014)
30. Suzuki, M., Yoshida, Y.: On the development and utility of action control individuality for semi-autonomous intelligent robots. In: 2019 18th European Control Conference (ECC). pp. 3550–3555. IEEE
31. Van Valin, R.D., van Valin Jr, R.D., van Valin Jr, R.D., LaPolla, R.J.: Syntax: Structure, meaning, and function. Cambridge University Press
32. Wehrmann, J., Cerri, R., Barros, R.: Hierarchical multi-label classification networks. In: International Conference on Machine Learning. pp. 5075–5084 (2018)
33. Weston, J., Chopra, S., Bordes, A.: Memory networks
34. Wu, T., Tjandrasuwita, M., Wu, Z., Yang, X., Liu, K., Sosič, R., Leskovec, J.: Zeroc: A neuro-symbolic model for zero-shot concept recognition and acquisition at inference time. arXiv preprint arXiv:2206.15049 (2022)
35. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. pp. 2048–2057 (2015)

36. Xu, R., Xiong, C., Chen, W., Corso, J.J.: Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In: AAAI. vol. 5, p. 6. Citeseer

4 Appendix - proposed DLM

4.1 Additional overview

In the following we present different aspects of the proposed DLM.

Multi-modal fusion There is evidence of this multi-modal fusion in the literature about image captioning or video recognition tasks. For example, a visual input is encoded into spatio-temporal space, and sentences describing the visual input are encoded into a continuous vector space. Then, the goal is to minimize the distance of the outputs of a deep visual model and a compositional language model in a joint space, and eventually to update these two models jointly [9,29,36]. We use this method in our proposed DLM, as discussed in 2.1.

Neuro-Symbolic perspective The joint space learning aims to produce symbolic representation. This idea is not new, in combining DL with symbolism, and referred to as the Neuro-Symbolic framework. It is expressed in many studies, e.g. Logical Neural Networks (LNNs) [26], the Neuro-Symbolic Concept Learner for Visual-Question-Answering tasks [22], and ZeroC [34] implementing composition of concepts.

DLM's components The inner components of the proposed DLM are replaceable, and can be implemented via appropriate DNNs. Sensory input can be handled by e.g. CNN, DBN, and SAE. Text input can be handled by e.g. Transformers, RNNs or their variants: LSTM or GRU. The Sensors-and-Text and the final decoder can also be implemented via sequence-based RNN, as in [1]. Because based on [12], the grasping of a situation is gradual in time. It takes time to figure out the stable situation, and it takes time to follow up on some desired plan. A plan is realized by a sequence of actions, such as in [30], where a robotic-complex-action is transformed, transferred, and then used for the control of base actions. The response, depicted in Fig. 1(b), can be either a physical operation or a sequence of words (e.g. an answer to a question).

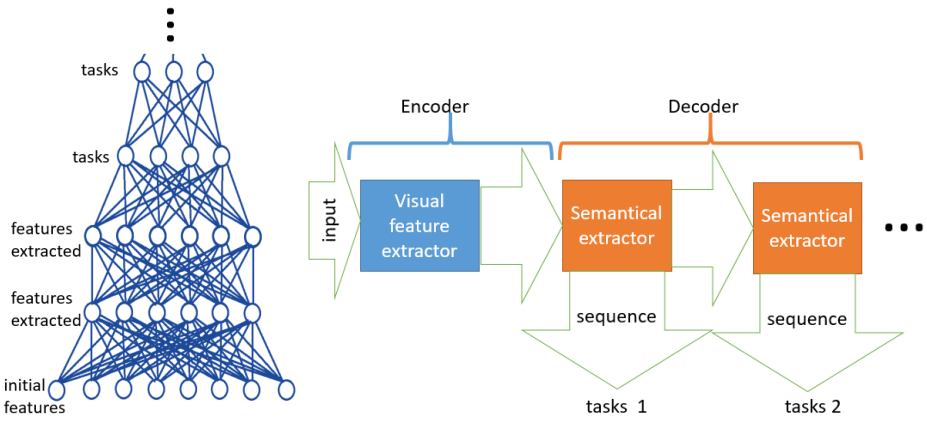
Hierarchical temporal structure The hierarchical temporal structure of our DLM can be implemented via different clock rates, as suggested in [14]. Another way is via sliding/shifted LSTM blocks as in [8], used to extract different time-scaled features. And another way is via dilated casual convolution, as in [10].

Gradual Learning The gradual learning we implement in our DLM also exist in literature in different forms. For example, in [4] gradual learning is proposed from a simple level to a complex level, either manually (expert-guided) or automatically (scoring each sample by its training loss). However, this loss is highly dependent on the models and their hyper-parameters. Hence, different learning takes place: from fewer categories or output tasks (local) to more categories (global).

Hierarchical Learning If we consider the hierarchical structure to be built as layers of features and tasks, see for example Fig. 7(a), then hierarchical learning can be done in the encoder as described in [3], e.g. in DNN or CNN. Also, features can be between tasks, as in [3].

But if our intention by "teaching" is matching words or phrases for the visual input, as in image captioning, a stationary type of learning, using the encoder-decoder structure, then we can separate tasks in different levels in the decoder, as illustrated in Fig. 7(b).

First, the model consists of visionary and semantic feature extractors, and we train on all tasks of group 1. Then we add another unit of semantic feature extractor to train on the composite tasks of group 1, which is group 2. And thus go on in the same manner.



(a) The output of each layer may be features or tasks

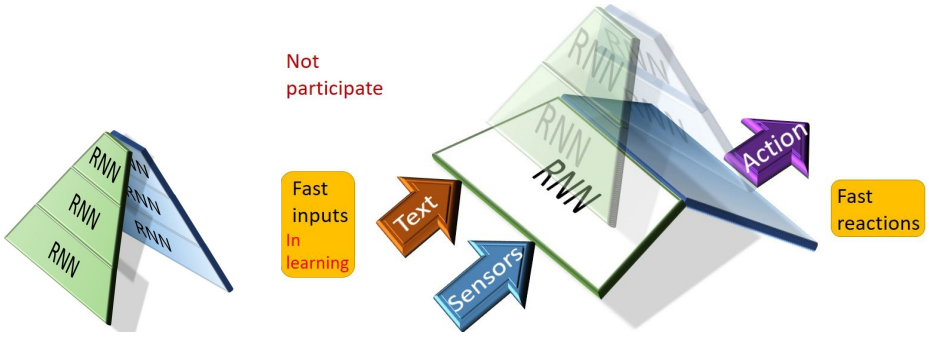
(b) Hierarchical learning in visual captioning

Fig. 7. Hierarchical learning

Spatial embedding prior to Spatial-temporal embedding Although we implement spatial-temporal joint embedding as our basic and first phase, it is possible also to check the option of adding joint embedding space for spatial information only, prior to the spatio-temporal short-term joint embedding, as it

is done for example with static visual images and simple textual objects [19]. This embedding enables the learning of static compositionality of objects, while later, the inclusion of temporal dimension enables temporal compositionality learning.

3D incremental learning version of the model Lastly, until now we have discussed gradual learning in the encoder of our DLM, after which we finish with supervised learning for the final response, via the decoder. Nevertheless, we can perform gradual learning also in the decoder. First we teach the encoder-decoder fast tasks with immediate execution. Next, we fix these first layers and teach the mid-term layers, and continue with the same fashion. This resembles the biology-based approach, in which after repeating some task, it becomes automatic for us such that our mind is free from concentrating on it, and now we can deal with other tasks while performing these low-level tasks. Similarly is here: after accomplishing the low-level tasks at a high level of performance, we are free to learn new tasks. This gradual learning can also introduce a working memory or thinking in higher available layers, where the inputs are very slow/stable, and allow the DLM to solve difficult tasks. This idea can be visualized more clearly in 3D, see in [16] or in Fig. 8.



(a) 3D view of Fig. 2 (b) Fast stimuli-response layer

(c) Slow stimuli-response layer

(d) Slower stimuli-response layer

(e) Slowest stimuli-response layer

Fig. 8. 3D incremental learning version of the model in 2.1.

Here we see an example of our hierarchical-temporal DLM, implemented in a traffic signal control problem. We utilize the text inputs as a command input channel.

In Fig. 8(b) we see fast commands. The input is the command and the state of the network. For example: minimum delay in the signalized junctions, with the scenario of a breakdown, or another command of avoiding queue spillbacks, in a regular network state.

In Fig. 8(c), a slower commands for this moment are mainly preference commands. For example to prefer some region in a regular network situation, or not to prefer some region cause it's in a breakdown or under construction works. How it is related to the lower-layers which are already-trained and nearly-fixed? First, it allows recognizing some long-term patterns, such as of forming congestion or the effects of an accident somewhere, then, acting with some manner that changes the action space for the fast tasks in lower layers. Meaning, that when lower tasks were alone, they had the largest freedom or the least constraints on their actions, such as full range over green times. But when higher tasks are operating, they decide upon a more restrictive actions available for the lower task executions. We also have skip connections for the lower level to allow it continue operating almost independently. Specifically the connection that fast tasks had in previous slide.

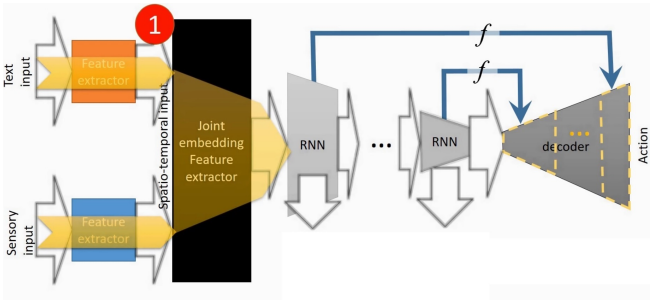
Next layers, see Fig. 8(d,e), operate in the same fashion. We can imagine it might even be used for thinking, or solving difficult tasks, that are to be transformed to the more fine solvers in the lower layers. How is it possible? Because the inputs to these layers are very slow/stable, so these layers don't need to process the fast tasks, thus they are left to deal with the other tasks given in the text input.

Training illustration We can illustrate how training occurs in the encoder via the second idea: visual data and equivalent or/and complementary textual data about objects and actions are merged in a joint space, see Fig. 9(a).

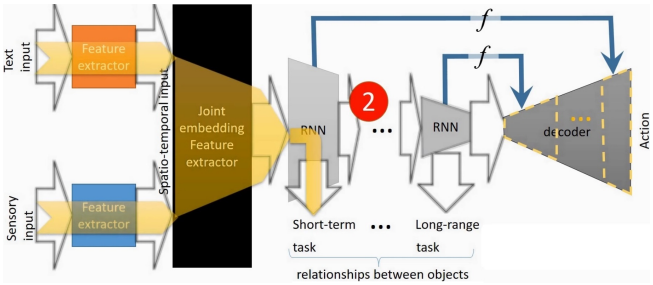
Then it traverses to one of the operating (short/mid/long-ranged) RNNs, to accomplish the task of predicting the scene, i.e. of the correct relationship between objects and actions in it, for example via graph representation (scene graph). It can be seen in Fig. 9(b-c).

Finally, a supervised learning is performed in the full DLM, specifically in the different layers of the encoder-decoder in the DLM, see Fig. 9(d-f).

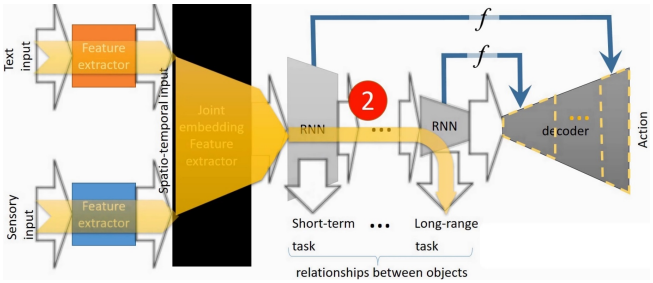
Full DLM training phases are in "DLM training" in [16] or in Fig. 9.



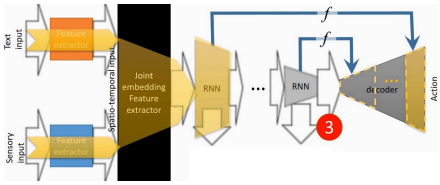
(a) Joint embedding learning



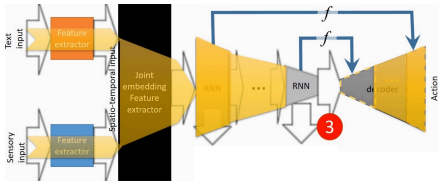
(b) Short-term encoding tasks



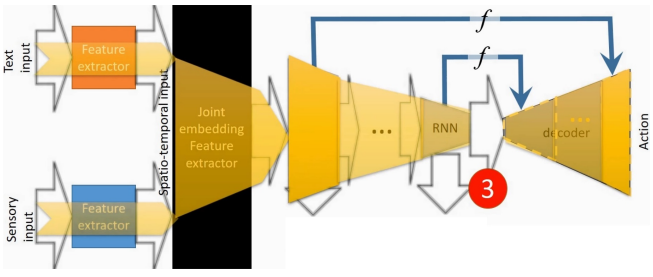
(c) Long-term encoding tasks



(d) Short-term decoding tasks



(e) Mid-term decoding tasks



(f) Long-term decoding tasks

Fig. 9. Full training illustration.

4.2 Additional suggestions for the proposed DLM

In the following we present different suggestions for the proposed DLM.

We should include a memory [33], either implicitly in the learning NN components themselves, or explicitly as additional components in the model, with a different type of memories, e.g. sensory, conceptual, and procedural memories.

Moreover, we could have a memory suited for the fusion of sensors with text, which should hold different temporarily categorized concepts. I.e. fast changing concepts and slow changing concepts, and all in between.

This system is compositional, and the modules in it are potentially learned separately. Hence, new objects/actions can be introduced by retraining the relevant modules.

A few additional aspects are presented for the proposed model. First, initially we thought to have a single channel for both informative text, describing the current situation, and instructional text, asking the system to perform something. But we figured we should separate these channels, due to several reasons: (i) When our text input is a command, then the NN is trained by using executions as outputs. However, there is no output to yield from a descriptive text. (ii) Often both channels are needed simultaneously: a descriptive one, such as coming from the user or some online source, and a commanding one. Furthermore, we presume that a commanding input represents our system's objective, and this objective has to be supplied consistently.

The second aspect is about a full-model training phase. After training on intermediate tasks to produce more representative features, we train the DLM on its actual output: the response (actions or answers). In this phase, we perform only feature extraction and disregard the intermediate tasks, since their function is needed no more.

4.3 Issues with the proposed DLM

In the following are some issues the proposed DLM has:

* The separation of different temporal scales into three discrete layers both in the encoder and the decoder are artificial, while it actually should be continuous.
* The issue of catastrophic learning exists here due to our batch learning setup. It means that we cannot go back to previous phases to add or update new concepts or relations, without ruining the whole system (or the other tasks).
* The DLM propose learning both simple and complex objects and their relations. There are DL studies that try to tackle the problem of modularity and compositionality of concepts [34], though there is still an issue to implement this. Hence, it is not straightforward that we can train the DLM to represent conceptual memories as depicted in Fig. 10.
  Similarly, the decoder in our DLM is a simple mapping from latent space to actions. But procedural memory, as it is perceived in AGI, requires symbolic representation and sequential processing, such as in RL. An additional constraint is causality in this representation.

Although our DLM aims to generate symbolic representation in the encoder, and then continue processing it in the decoder, it is still not guaranteed for the features above to be realized. Nevertheless, sequential processing or planning is designed in the hierarchical temporal structure of the decoder. It supposes to be hierarchical planning, starting from slow higher levels utpo the fastest lowest levels.

* AGI should reflect logic, reasoning, planning, and other constructions [18]. However, the proposed DLM, in its current state, does not possess these functions, though it allows for timeless processing at its higher levels. It is where the processing is so slow, that we almost disregard the inputs/outputs. Moreover, DL is limited in various AGI necessary capabilities, such as adaptation and generalization (especially to out-of-distribution) learning speed, knowledge transfer, reliability in reasoning tasks, and more.
  Additionally, learning in its wider view, is not only algorithmic, but is also rule-based and can be unified with reasoning. It also has many other forms, such as: reinforcement, imitation, and instruction.
* The proposed DLM is an open-loop system, trained via supervised learning, i.e. it has no feedback from the environment, which is a significant limitation, especially when interacting with the environment. Though it could simply close the loop and provide some reward, thus transferring it to a Reinforcement Learning (RL) regime, which has a better potential to achieving AGI. This vision is supported both in the AGI community [2] and in the DL community [23].

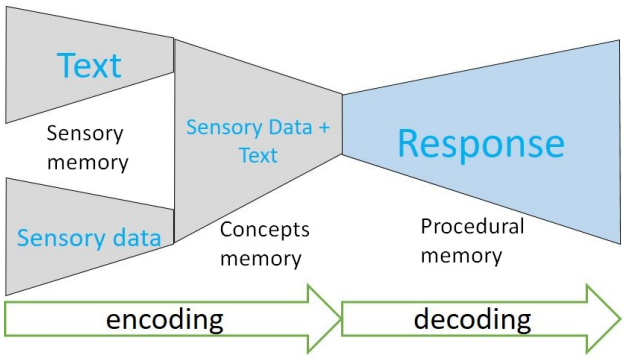


Fig. 10. Different memory representations in the proposed DLM.

Some of these issues can be dealt with by several optional solutions. But it will not be addressed here.

5 Appendix - Notes about AKREM

Additional notes about AKREM are presented in the following.

5.1 Relation to NLP/NLU

Two comments in related to Natural Language Processing (NLP) or Natural Language Understanding (NLU).

Firstly, as one can see, AKREM is an alternative to Noam Chomsky's formal language [5], specifically the language parsing. On the other hand, it is similar to the meaning-based parsing in [31], by also bypassing the need for semantics to be founded upon syntax parsing, i.e. upon Parts-Of-Speech (POS).

Secondly, AKREM's story-like modeling of sequential 1D input demonstrates the deficiency of RNNs and LSTMs and the success of transformers. It illustrates how we construct a full model from a sequential input, considering all the sequence (attention), not only the last parts of it or its summary (RNN/LSTM).

5.2 Retrieval

It is a very comfortable representation for a multi-use retrieval, i.e. its structure allows for retrieving either the whole story sequentially or parts of it by demand.

It could be either purposeless retrieval, usual reminding, or it could be purposeful one, e.g. when having a problem to solve, such as answering a question(s).

For example, in the video in [link], if one asks "What was David doing in his room?", the agent starts at some point in the hierarchy, and moves along its highest level, to find the most relevant node, then after finding one, descends to search similarly in the lower level, and so on.

However, it can be a much more complex trajectory. For example, when being asked "What did David do after his mom called him?", it first searches for the point when mom called him, so that it can look for what he did afterward. I.e., it is a multi-hop type of trajectory. It can go up and down as much as needed.

5.3 Motivation

Based on the retrieval function presented above, we present our original motivation for AGI research.

As coming from an optimal control field originally, we encountered an inefficient optimization over datasets. It means, that every new task is compelled to rerun a given or a new optimization algorithm. Even when moving to the AI field, specifically DL, it stays the same problem, though extended a little bit.

For example, multi-tasking allows for multiple tasks in one learning period/shot, but the tasks must be pre-defined. Similarly in meta-learning, where the tasks are grouped into training and testing sets.

Differently, language models in NLP try to address this issue, especially transformer-based ones which use prompting as input. In this case, we can insert the question with the data, to "attack" this data from different directions, to test comprehension.

The main motivation is that the AGI agent should learn the data efficiently, at a level of "understanding", such that the learned data is learned in a most-usable manner, i.e. to be used for different tasks, especially for tasks that were not learned in the given data context.

We can imagine it like a student in an English lesson. He is reading a story, and then performing a comprehension test afterward. This test should both examine the understanding of the story-only facts, but also in a wider context, such as a part of previous knowledge and common sense.

This motivation, hopefully, presents AKREM in a different light, where a multi-usable retrieval should supply the need for the ultimate understanding of a story/message.

5.4 Limitations

The current version of AKREM obviously has various limitations, specifically from the practical implementation perspective, since it is a preliminary idea.

Here are some of these limitations:

* AKREM is not representing cognitive processes, like learning and adapting. However, this representation facilitates functions like memory generation and retrieval, perception (recognition), reasoning, planning, knowledge transfer (e.g. via analogy), causality, and more.
* Representation by itself does not provide the means to derive it. Meaning, how the grouping and separation occur in the hierarchy is a subject for discussion, and is not being formalized yet. Similarly, there is no mechanism for retrieval in stored hierarchies.
* AKREM representing instances of events/entities, i.e. it does not allow for abstraction into general classes. The next paper includes this feature in its model.

5.5 Contribution

It is a raw idea, and lacks the accurate mathematical formulation and the experimental evidence for proof of concept.

However, it suggests a novel representation approach, based on associative thinking, thus proposing a compact representation of context. It supports human natural (and not structured) communication, thus addressing also memory generation and retrieval. Accordingly, it deals with memory organization, i.e. of LTM and WM.

It also tries to facilitate as many different cognitive aspects as possible. Aspects such as perception (recognition), reasoning, planning, knowledge transfer (e.g. via analogy), causality, and more.

Finally, it inspires further discussion and revision of current NLP paradigms, such as POS parsing. Especially, in case it will become a practical interest, a serious discussion should be conducted about realizing will and purposefulness in language representation.

6 Appendix - Generalization in AGI

Generalization interestingly can occur by somehow abstracting out different contexts and grouping the commonalities. For example, when one sees dogs in different circumstances, and for each one of them he is being told that it is a dog, then he connects all these events together, to learn some operational characterization: they have attributes like fur, small bodies, and their unique behavior. Similarly, we learn math by abstracting out the specifics of the many examples we learn, left out eventually with an exact algorithm for doing math. Furthermore, in any skill and action, we can generalize beyond some specific object, to perform the same series of actions over other objects as well. This is referred to as analogy or transfer of knowledge.

Hence, humans prefer a rule-based approach, since it encapsulates many scenarios, instead of low-level specific examples, as in DL. This may explain why we are drawn naturally to the rule-based approach (since it is actually the result of abstraction).

If we combine this AGI characteristic with our need to model everything we interact with (see 3.1), we come up with one possible insight. We need some sort of consistent reorganization of previous data, to turn it into abstract models, on which we can perform predictions. Anything else, which is not modeled, is not assigned for prediction. Models are the most efficient knowledge representation, since beyond prediction they can also simulate different scenarios, e.g. answering questions, understanding different aspects of a concept, and applying counterfactuals.

Note: the discussion above is a promo for the next paper, about a model of models.