Study on Chunking Mechanisms for a 3-layered Associative Memory and Recall Model

Mungai, Peter Kimani

(出版者 / Publisher)
法政大学大学院情報科学研究科
(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 情報科学研究科編
(巻 / Volume)
14
(開始ページ / Start Page)
1
(終了ページ / End Page)
6
(発行年 / Year)
2019-03-31
(URL)
https://doi.org/10.15002/00021927

Study on Chunking Mechanisms for a 3-layered Associative Memory and Recall Model

Peter Kimani Mungai
Graduate School of Computer and Information Sciences
Hosei University
Tokyo, Japan
mungai.peter.69@stu.hosei.ac.jp

Abstract—Memory models are important components that support AI computational models to learn and remember things. Usually, computers operate using an address-based memory for storage where each memory item resides in a unique location. This study proposes a 3-layered associative memory model based on chunking mechanisms of the brain to store knowledge in form of association between entities. The human brain is a cognitive model that derives information from sensory data like vision, auditory, and touch, associates different patterns to create knowledge and uses chunking mechanisms to package the acquired knowledge into manageable entities. Through Chunking, each item held in the STM is a singular entity (chunk) containing more associations (knowledge) in it. Using chunking mechanisms of the brain, AMR model can store knowledge in manner that enables faster response to stimuli. To represent knowledge and semantic relations effectively, the hyper structure (concept) defined in denotational mathematics is employed. A network of concepts is maintained in a cognitive knowledge base that continually evolves as knowledge accumulates. The chunking mechanisms used in this study are goal-oriented chunking and automatic chunking.

Keywords— Memory models; associative learning; Hopfield model; chunking mechanisms; short term memory; Long term memory

I. Introduction

In the modern world, artificial intelligence technology has become ubiquitous. Many people have become dependent on AI powered computer applications to perform their day to day activities. Typical daily activities such as communication, transport, healthy living and education are now mediated by computer technology. This unprecedented growth of AI can be largely attributed to the affordability of powerful devices such as smart phones and personal computers. As the world moves to an era of the internet of things (IoT), mobile devices and other smaller devices will be expected to run complex software systems efficiently. This is largely because users will expect these devices to become their dependable and long term cognitive partners. As such, finding ways of utilizing minimal resources to facilitate learning and recall is paramount in the IoT world.

A major hurdle to be overcome before general AI can become a reality is to have intelligent models that are able to learn and recall different tasks [1]. This can be only achieved if such models have access to an efficient memory and recall mechanism. In the case of neural based models, they need to learn and represent a wide array of knowledge in their weights. Most neural models can only handle a limited number of tasks reliably because learning of new tasks normally requires alterations of the weights associated with previous tasks. This alteration of weights associated with previous tasks when learning new ones in neural networks eventually leads to the problem of catastrophic forgetting. For other AI models like the rule-based systems, they become increasingly complex as more rules are added to the knowledge base eventually leading to conflicts making them ineffective. The future success of intelligent models will depend on how efficiently they will be able to learn and recall different kinds of tasks. The existing models of learning and recalling knowledge might not be able to cope with such a requirement based on their design. In this study, we look for inspirations from biology, where researchers have discovered that the mammalian brain is able to perform its cognitive functions efficiently while using little resources compared to the ordinary computer models.

The human brain being the command center for the human nervous system contains about 86 billion nerve cells (neurons) [2]. These nerve cells link to each other through axons and dendrites and are believed to be the basic memory units. During stimulation, the neurons influence each other to either fire or not fire (transmit an electrochemical signal). This binary nature of how the neurons behave can be easily simulated in a computer model. The gist of the matter however lies in how memories are formed and recalled by the brain. Psychologists have for long held that humans learn and remember through associating often unrelated items through a process known as chunking. Recent empirical evidence from experiments involving observation of experts seem to support this claim.

Chunking is a kind of cognitive compression mechanism where the brain parses information into sub-components that are more memorable and easier to process than the seemingly random bits of which they're composed [3]. For instance, humans learn and recall long sequences of phone numbers by segmenting the sequence into small segments. E.g. a phone number 0724942245 would be easier to remember when divided into 3 segments; 0724-942-245. Chunking is the hallmark of the brain's organization [4]. Numerous psychological experiments have confirmed that experts in given domains accumulate their expertise through chunking. Chess masters acquire skills when constellations of pieces (segments of the game) become associated with moves or strategies and are stored in the long-term memory [5]. It rightly appears that the role of chunking is

Supervisor: Prof. Runhe Huang

to move to a given goal through divide and conquer principle. This principle is vivid in most computer paradigms such as object-oriented programming where larger problems are solved by solving sub-problems. These realities make chunking a practical approach when building a computer model as well as an effective strategy of improving learning in cognitive learning models.

II. Problem Description

This study is anchored on trying to answer the questions; (1) why artificial intelligent models are not as efficient as the human brain in learning and remembering a variety of tasks, (2) why the human brain despite its low capacity short-term memory, it can handle a variety of tasks almost simultaneously and (3) how we make a learning and recall system that can run on a small IoT device to offer reach human-machine interactions. To Answer these questions effectively, this study departs from the traditional design of memory models by creating an associative memory and recall (AMR) model that mirrors the human brain's cognitive functions of learning and recalling. Creating an efficient memory and recall model will help expand the application domain of smart devices as they will be in a position to handle different scenarios. Furthermore, the proposed model will support the deployment of highly adaptive systems on small devices by efficiently storing and recalling the knowledge they require to operate. The use of chunking mechanisms as a main component of the proposed AMR model augers well with the attempt by most researcher to create human centric systems that become cognitive partners to human beings. The mathematical model that makes up a chunk is the notion hyper structures in concept algebra and denotational mathematics given in (1) that is inspired by [6,7].

$$H=(E, A, R^{i}, R^{\ll}, R^{\gg}) \qquad (1)$$

Where, H is hyper structure made up of a finite set of entities E, a finite set of properties A, a finite set of internal relations R^{i}, a finite set of prior relations $R^{\ll}$ and a finite set of post relations $R^{\gg}$. A chunk can be made up of one or more hyper structures. Relationships between different chunks can be accurately captured in a Hopfield network to create an associative memory. The overall operations required to record new knowledge or retrieve memory will also be reasonably minimal for the AMR model to be biologically plausible i.e. to respond to a stimulus, the AMR model should not simply compare thousands of possible responses like most computational models but rather sort its memory in manner that allows easy remembrance and storage just as is the case with the human brain.

III. Proposed 3-layered AMR Model

The proposed model is a 3-layered memory structure that is focused on efficiently storing knowledge using chunking mechanisms. A single chunk can hold complex association of entities representing knowledge therefore acting like a rich package of information. The proposed model contains 3 layers i.e. *learning*, *operations* and *semantics layers* and uses associative memory to store acquired knowledge. The stored knowledge is in hierarchical order for easy use by an effector in the environment. Fig. 1 shows an overview of the proposed AMR model.

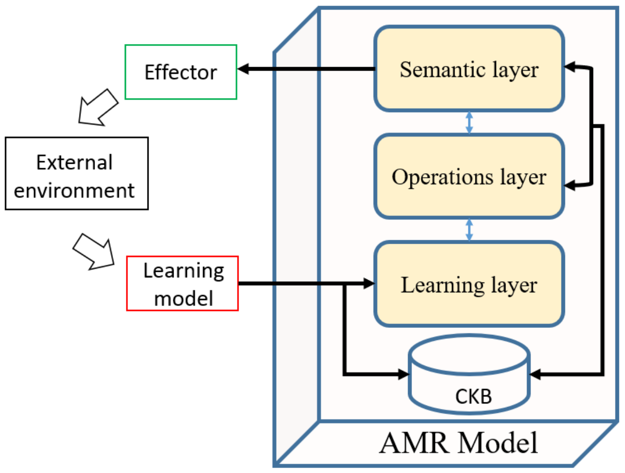

Fig. 1. The Overview of the 3-layered AMR Model

A. The Learning Layer

The learning layer uses a Hopfield neural network to learn an input pattern. The nodes in a Hopfield neural network influence each other's states until they get to a stable state. A Hopfield neural network accepts a (*1 x n*) pattern and uses a learning algorithm like the Hebbian rule given in (2) to create an (*n x n*) weight matrix. The Hebbian learning rule defines how to calculate the weight between two interconnected nodes i, j.

$$w_{ij} = \frac{1}{p}\sum_{k=1}^{p} x_i^k x_j^k \,, \quad i \neq j \qquad (2)$$

Where, w_{ij} is the synaptic weight between node i and j, p is the total number of training patterns and x is a bit in the k^{th} pattern. For a Hopfield neural network with n nodes, there can be $n-1$ such equations which make up the Hopfield weight matrix. Fig. 2 shows a schematic view of Hopfield learning in the AMR model. A conceptualizer holds concepts data in form of a hash table from the knowledge base. A real-world fact is mapped to its corresponding representations in a Hopfield network. There are other learning models that can also be used to generate the knowledge pattern vector such as the KID model.

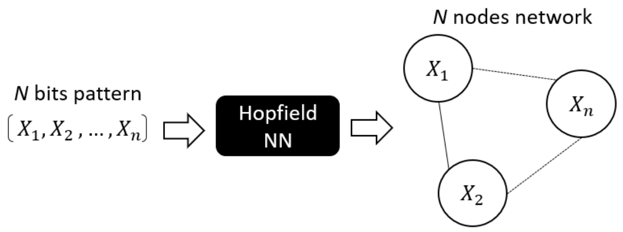

Fig. 2. Hopfield Learning in AMR Model

B. The Operations Layer

The key goal of the operations layer is to find associations (if any exists) between the input matrix and other matrices in the repository. In cases where matrices are found to be associated they are combined using merging mechanisms. Merging of related matrices can be achieved through several algebraic means namely; singular value decomposition (SVD), lower triangular and upper triangular (LU) decomposition and matrix expansion method. Before two matrices can be marked for merging, they should be similar. Algebraically, two matrices *A* and *B* are similar if they satisfy (3) where *P* is another matrix. *A* and *B* should also have the same rank, determinant and Eigenvalues.

$$B = P^{-1}\, A\, P \tag{3}$$

When using the SVD method, the input weight matrix is factorized into submatrices. Secondly, the sub-matrices are checked for similarity by querying the knowledge repository for any potential associates and comparing them recursively. As such this operation returns an array of all matrices associated to the current input. Thirdly, the header vector is updated with the list of associations returned by the similarity checker method. Lastly, the merging method is invoked to create new matrices based on the associations of the input matrix. The merging process is based on reverse SVD whereby associated weight matrices are treated as sub-matrices that yield the main matrix using the SVD formula. The new matrices are then added to the knowledge repository. The SVD formula is shown in (4) where, *M* is the main matrix while *U*, *J* and V^T are sub matrices.

$$M = U\, J\, V^T \tag{4}$$

The LU decomposition requires the sub-matrices to be opposite triangle matrices for it to work. Matrix enlarging involves creating a large matrix that contains the elements from all the sub-matrices by simply adding new nodes to the Hopfield network. The process of creating a new matrix from its constituent sub-matrices and establishing the connection strength between the nodes is also known as automatic chunking and will be discussed on later parts of this paper.

C. Semantic Layer

The main goal of the semantics layer is to create and update a concept network that can give an elaborate inference about the input Hopfield weight matrix. The elaborate inference is in form of a list of all the associations related to the new matrix. This layer is primarily made up of a semantic network together with an object-oriented representation approach. To construct a semantic network, the module loads activated concepts in the knowledge repository using goal-oriented chunking mechanism with their accompanying header vectors that indicate their relationships. Based on the header vector values, the semantic module creates concepts and their sub-concepts. Each weight matrix becomes an instance of a given concept or sub-concept in the semantic net as shown in Fig. 3. The goal of the semantic module, therefore, is to assign a position to the new weight matrix within the network. Based on the position assigned, an elaborate inference of concept associations is outputted by the module which acts as the simulation result of the overall system.

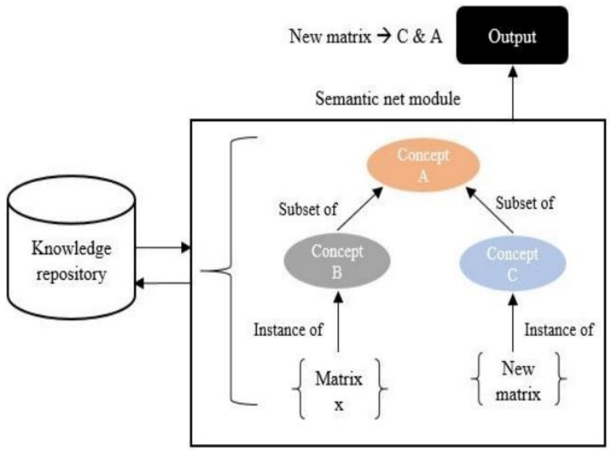

Fig. 3. Semantic Network Generation

IV. Chunking Mechanisms

Chunking initially proposed by De Groot [8,9], is believed to be critical for learning and cognition both in humans and animals. According to Miller [10], a chunk collects several pieces of information from the environment into a single source. There two types of chunking namely; (1) goal-oriented chunking which is a conscious process and (2) perceptual chunking which is an automatic and continuous process [8]. Chunking enables the brain to overcome memory capacity limitations especially in the short-term memory by grouping together related items. According to Gobet and Simon [8], frequently used chunks become templates. Templates are what make experts tick since they can retrieve information from a template quicker compared to a normal chunk. The use of chunks explains how greater knowledge can lead to an increased ability to extract information from the environment despite the constant cognitive limitations. The idea of chunking and templating to compress information would be very useful for computer-based model where efficiency of training and recall is key.

A. Automatic Chunking

This chunking is based on the theories put forward by de Groot [8] and Chase and Simon. According to these theories, the entities within a chunk are bound by strong association links. Expressing a chunk using denotational mathematics and concept algebra, a chunk's (formal concept) internal entities are its entities and properties. The cosine similarity used to find similarities between entities and properties. The cosine similarity between two vectors $\vec{a}$ and $\vec{b}$ is a measure of the cosine of the angle θ between them i.e.:

$$\cos\theta = \frac{\vec{a}.\vec{b}}{\|\vec{a}\|\|\vec{b}\|} \tag{5}$$

Where, $\vec{a}$ is an attribute of a given formal concept or chunk *C*, $\vec{b}$ is an object of the same formal concept or chunk C, and $\cos\theta$ is the angle between the attribute vector and the object vectors $\vec{a}$ and $\vec{b}$.

The smaller the angle θ between an entity and a property, the similar they both are. Algorithm in Fig. 4 shows the automatic chunking process in the proposed model. The algorithm can also be configured on line 3 to chunk opposite objects or dissimilar object. This allows for derivation of rich semantic relation between chunks just as the human brain does. The cosine similarity model is effective since the angle θ between objects can rigorously establish their relationship i.e. similar if $0 < \theta < 90$, dissimilar if $90 < \theta < 180$ and opposite if $180 < \theta < 360$.

```
Algorithm 2: Automatic Chunking
  Input: Concept
  Output: Concept{internal relations}

1. FOR ALL entities & properties IN concept
2.     CALC cosine similarity $\theta$
3.         IF $\theta$ > threshold
4.             $R^*$ += entity🡸🡺property
5. Return concept{$R^*$}
```

Fig. 4. The Automatic Chunking Algorithm

B. *Goal-oriented chunking*

This chunking mechanism is based on the theory of Miller [10] who held that chunking is a deliberate process actively undertaken by a chess player to build expertise. Since the knowledge to solve a problem might reside in different chunks, such an expert would have to devise a way of associating them. This study simulates goal-oriented chunking by finding inter chunks associations. In the brain, inter chunk associations might be helpful in ensuring an activated chunk is loaded to the short-term memory together with its closest associates. In this way, the brain can efficiently carry out tasks without having to invoke the long-term memory.

Since an activity is formally represented in this study, its association with other activities can be discovered by analyzing the external relations. All associates of a Formal concept can be found according to (6).

$$H^*: \forall_m = \sum_{a=1}^{n} H_a\{E_a, A_a, R_a^i, R_a^{\ll}, R_a^{\gg}\};\ \ R_*^{\ll} \approx R_a^{\ll}, R_*^{\gg} \approx R_a^{\gg} \qquad (6)$$

Where H^* is a concept (chunk) whose related concepts (chunks) are being sort, *n* is the total number of concepts (chunks) available for comparisons, H_a is the concept being currently compared with the H^*, E_a the finite set of entities of the activity H_a, A_a is the finite set of properties of activity H_a, R_a^i is the finite set of internal relations of activity H_a, $R_a^{\ll}$ is the finite set of input relations of the concept H_a, $R_a^{\gg}$ is the finite set of output relations of the concept H_a, $R_*^{\ll}$ is the input relations, $R_*^{\gg}$ is the output relations of the concept H_a, and $R_a^{\ll}$ is the input relations of the concept H_a, $R_a^{\gg}$ output relations of the activity H_a. A concept can be chunked in the AMR model using the goal-oriented scheme. This is done according to *Algorithm 2* in Fig. 5.

```
Algorithm 2: Goal-oriented Chunking
  Input: CKB {instance}
  Output: Concepts {external relations}

1. FOR concept IN {CKB}
2.     COMPARE stimulant AND concept
3.         IF stimulant_ concept ∈ Formal_ concept
4.             ASSOCIATE stimulant & concept
5. Return concept{$R^{\ll\gg}$}
```

Fig. 5. The Goal-oriented Chunking Algorithm

V. Related Work

A. *Memory models for preventing catastrophic forgetting*

Catastrophic forgetting [13] refers to a situation where a neural network will lose weights associated with learned previous tasks. To solve this problem, there are two models Elastic weight consolidation (EWC) model proposed by K. James et al [14] and differential neural computer (DNC) model proposed by A. Gravels et al [15]. This study compares the performance of the EWC model to the proposed model. Elastic weight consolidation (EWC) borrows heavily from the inner workings of a mammalian brain which was discovered to be retaining new skills through the strengthening of the synapses associated with that acquired skill. To perform a given task A for examples, weights were preserved as much as possible during the learning of another task B. The EWC model works by adjusting its weights when learning a new task along a direction where the previous task plane overlaps with the current task plane.

B. *Rule-based systems*

A rule-based system uses a set of productions to represent knowledge coded into the system [11,12]. Rule-based systems have been used to store knowledge and perform reasoning, but they suffer from two main challenges i.e. conflicting rules and complexity of the rule base when there is a dramatic increase in rules. Conflicting rules arises when one of the fired rules implies a certain fact while another rule negates the same fact. These challenges have a negative impact on performance and may render the system too slow.

VI. Human Activity Recognition Scenario

An experiment was conducted by F. Javier et al [16] to create an open data set on activities of daily living (ADLs). The data was based on the sensors fitted in the participants homes that wirelessly transmitted their values periodically. For each activity, its *start time*, *end time* and *activity name* were recorded. The established human activities were *Leaving*, *Toileting*, *Showering*, *Sleeping*, Breakfast, *Lunch*, *Dinner*, *Snack*, *Spare time/ TV* and *Grooming*. The distribution of these

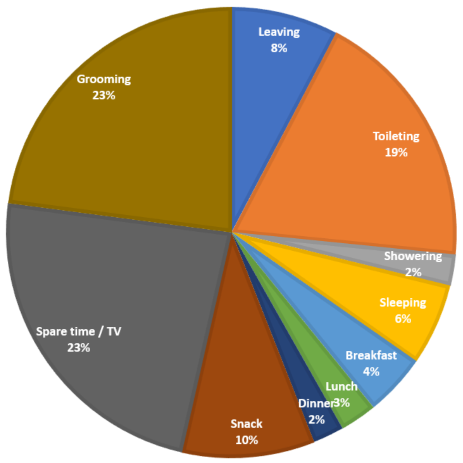

Fig. 6. Human Activity Distribution in the ADL Dataset

activities is as shown and labelled in Fig. 6.

The activities were inferred by human experts after analyzing data from 5 kinds of sensors fitted in the participants homes i.e. passive infrared sensor (PIR), magnetic sensor, flush sensor, pressure sensor and electric sensor. In addition to sending sensed values, the sensors also transmitted their IDs and locations. As such the human experts could determine the kind of sensor and its location inside the home from the data it transmitted. For example, when the electrical sensor attached to the TV transmitted its data, it was in the following form (ID: ***elec_sensor001***, LOC: ***living_room***, DATA: ***{true/ false}***). A human activity was formed by observing data from one or more sensors and associated duration. For example, watching tv activity is inferred when both the TV electric sensor and the seat pressure sensor sends true values i.e. (ID: ***elec_sensor001***, LOC: ***living_room***, DATA: {***true***}) and (ID: ***press_sensor001***, LOC: ***living_room***, DATA: {***true***}) respectively.

A. *Data Preparation*

An activity was encoded into a vector pattern based on sensors that form it since the AMR model accepts knowledge as a vector pattern. Table I shows the vectorized activities of the ADL dataset. This information is also used to create a concept look-up in the learning layer. The location IDs of the sensors were used to form the entity set of the activity concept while the sensor values formed the property set.

TABLE I. Activity Encoding to Vector Pattern

	Activity vectorization	Activity pattern
Leaving	*(main door[1], electric[0], pressure[0])*	*(1,0,0)*
Toileting	*(flush[1], shower[0])*	*(1,0)*
Showering	*(flush[0], shower[1])*	*(0,1)*
Sleeping	*(bed[1],electric[0], maindoor[0])*	*(1,0,0)*
Breakfast	*(fridge[1], toaster [1], seat [1])*	*(1,1,1)*
Lunch	*(cooktop[1], fridge[1], microwave[1])*	*(1,1,1)*
Dinner	*(cooktop[1],fridge[1],microwave[1],door[0])*	*(1,1,1,0)*
Snack	*(seat [1], fridge[1], microwave[0])*	*(1,1,0)*
TV	*(TV[1], seat[1], main door[0])*	*(1,1,0)*
Grooming	*(bed[1], cupboard[1])*	*(1,1)*

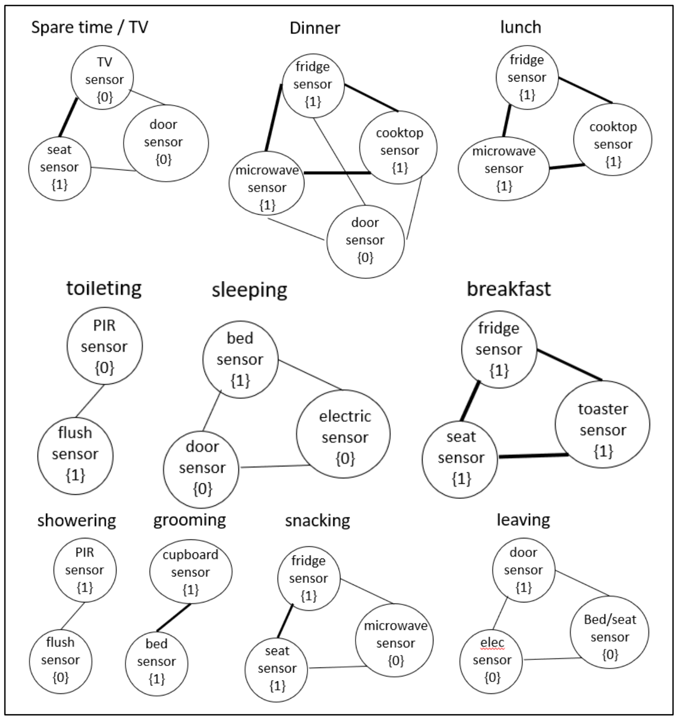

Fig. 7. Hopfield Networks Formed after Training

B. *AMR Model Training*

The encoded data i.e. activity pattern was used to train the learning layer's Hopfield network while the vectorization metadata i.e. names of the sensors making an activity were used to create a hash table that mapped a pattern to a concept (activity name). For example, to learn the activity TV which is represented by the pattern (1, 1, 0), the learning results of the Hopfield neural network using the Hebbian learning rule was as follows:

$$(1,1,0)^{T} * (1,\ 1,\ 0) = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad (7)$$

The resulting weight matrix store the activity pattern. Using the conceptualize module of the learning layer, the matrix is mapped to its properties (names of sensors that form it) as well the actual concept it represents (activity name) to form a Hopfield network.

C. *AMR Training Results*

The Hopfield networks for all the 10 activities were formed and held at the operations layer while their corresponding concepts or activity names were stored in the knowledge repository. Fig. 7. shows the resulting Hopfield networks after the AMR model was trained on all training set. The thickness of the edge between nodes indicates the strength of associations. The Hopfield networks' corresponding concepts are stored in the cognitive knowledge base.

D. *Scenario*

With a fully trained AMR model, knowledge about typical home activities scenarios can be tested. Using the test set data, different activities can be activated and their semantic connection to each other discovered. 3 consecutive activities in the test set data were selected to demonstrate how well the system would answer the questions *what*, *when* and *where*. After encoding the activities into vector patterns, the AMR model learning layer generated a Hopfield matrix that corresponded to the showering, grooming, snack and leaving concepts in the cognitive knowledge base thereby activating them. These activated concepts were used to form a semantic net in the semantic layer. The corresponding Hopfield networks in the operations layer were linked together to mirror the knowledge in the semantic layer. In this study, the sensor location IDs are considered entities while their values are considered properties.

E. Remarks

Chunking process mirrors the cognitive functionality of the brain where associated memory items are chunked together so that they can be loaded into the short-term memory as a single unit. The representation of activities as formal concepts allows for description of each activities internal and external associations. When compared to other models in this study, the AMR model can find the highest number of associations among activities without human intervention. Table II shows the comparison between the AMR model and rule-based system and the EWC model. To prove this claim, the Sussex Huawei activity recognition open dataset, was used to compare the long-term evolution of the AMR model compared to a rule-based system and the EWC neural network. The results were as shown in Fig. 8.

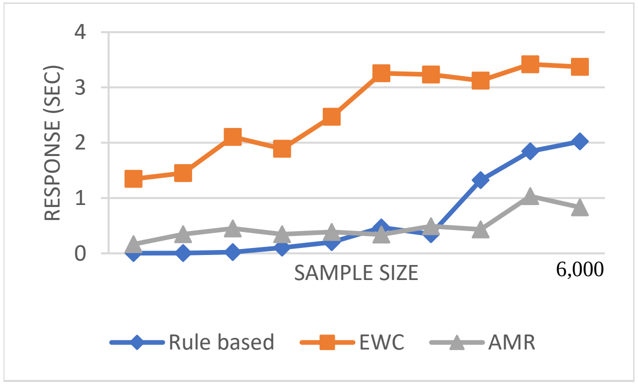

Fig. 8. Performance Comparison

TABLE II. AMR Evaluation

	Rule based	EWC model	AMR
Execution time	0.003	2.103	0.164
Associations	30	28	34

VII. Conclusion and Future Work

This study was aimed at creating a memory structure that mimics the internal working of the brain. To ensure memory items could be easily associated, they were formally expressed using the formal rules in denotational mathematics. A formal activity constitutes of internal relations, input relations and external relations. With these kinds of relations defined in an activity, chunking mechanisms could be applied to group associated activities together. Chunking allowed for packaging of related activities as a single entity. This is biologically plausible since chunking and templating theory of knowledge science show that experts can package information this way. Furthermore, the semantic layer of the AMR model acted as the short-term memory while the cognitive knowledge base (CKB) acted as the long-term memory (LTM). Since the execution time of the proposed model was quite high, we believe the increased number of associations discovered compared to other studies justify it.

References

[1] S. Legg and M. Hutter, A definition of machine intelligence, 3rd ed., vol. 521. Universal intelligence, 2007, pp. 391-444.

[2] T. Lewis, S. Writer, ‘Human Brain: Facts, Functions & Anatomy’, March 2016.[Online]Available:https://www.livescience.com/29365-human-brain.html. [Accessed: 23- Jan- 2018].

[3] M. Popova, ‘The Science of “Chunking,” Working Memory, and How Pattern Recognition Fuels Creativity’. [Online]. Available: https://www.brainpickings.org/2012/09/04/the-ravenous-brain-daniel-bor/. [Accessed: 12- Dec- 2017].

[4] J. Fonollosa, E. Neftci and M. Rabinovich, “Learning of Chunking Sequences in Cognition and Behavior,” PLoS Comput Biol 11(11): e1004592.doi:10.1371/journal.pcbi.1004592,USA 2015, pp. 1–24.

[5] A. Cook, "Computational Chunking in Chess," Ph.D. dissertation, Dept. Comp. Scie., University of Birmingham, Birmingham, England, 2011.

[6] M. Valipour and Y. Wang, “Formal Properties and Rules of Concept Algebra”,International Conference on Cognitive & Computing,IEEE, Beijing, China, July 2015 pp 1-8

[7] Y. Wang, “In Search of Denotational Mathematics: Novel mathematics Means for contemporary Intelligence, Brain, and Knowledge Sciences”, *Journal of Advanced Mathematics and Applications,* vol. 1, 4-25, 2012

[8] A. D. de Groot, Het denken van den schaker, Noord-Hollandsche Uitgevers-Maatschappij, Amsterdam, 1946.

[9] A. D. de Groot, Thought and Choice in Chess, Noord-Hollandsche Uitgevers-Maatschappij, Mouton Publishers, 1978.

[10] G. A. Miller, (1956) The magical number seven,plus or minus two: some limits on our capacity for processing information. Psychol. Rev. 63, 81–97.

[11] A. Ligęza. “Logical Foundations for Rule-Based Systems”, vol.ku 0146. Scientific Publishers of AGH-UST, pp 149-185 2005.

[12] J. Durkin. “Expert System: Catalog of Applications. 1st Edn., Intelligent Computer Systems.” Inc., Akron, OH., ISBN 0-12-670553-7 (1993).

[13] J. Kirkpatrick et al, “Overcoming catastrophic forgetting in neural networks,” arXiv. 1612.00796 [cs.LG], London, UK, 2017.

[14] K. James et al,1 Overcoming Catastrophic Forgetting in Neural Networks, 3rd ed., vol. 2. National Academy of Sciences of the United States of America, Feb 13, 2017, pp.1-6.

[15] A. Graves et al. Hybrid Computing using a Neural Network with Dynamic External Memory, International Weekly Journal of Science, Oct 27 2016, pp 1-21.

[16] UCI.com, Activities of Daily Living (ADLs) Recognition Using Binary Sensors_Data_Set.[Online].Available:https://archive.ics.uci.edu/ml/datasets/Activities+of+Daily+Living+(ADLs)+Recognition+Using+Binary+Sensors. [Accessed: 02- Feb- 2018].

[17] A. Sato, R. Huang, From Data to Knowledge: A Cognitive Approach to Retail Business Intelligence, IEEE International Conference on Data Science and Data Intensive Systems, Sydney, NSW, Australia, Dec 2015, pp 1-8.