5-2017

A Comparative Study of Cognitive Systems for Learning

Praneetha Mandava

A COMPARATIVE STUDY OF COGNITIVE SYSTEMS FOR LEARNING

Praneetha Mandava

Columbus State University

The D. Abbott Turner College of Business

The Graduate Program in Applied Computer Science

A Comparative Study of Cognitive Systems for Learning

A Thesis in

Applied Computer Science

by

Praneetha Mandava

Submitted in Partial Fulfillment
of The Requirements
for The Degree of

Master of Science

May 2017

ABSTRACT

Learning is the modification of a behavioral tendency by experience. Memory and reasoning are the most important aspects for learning in humans; information is temporarily stored in the short-term memory and processed, compared with existing memories and stored in long-term memory, and can be re-used when needed. One way to describe an organized pattern of thought or behavior and the categories of information along with their relationships is by using schemas. A cognitive script is one form of a schema that evolves over multiple exposures to the same set of stimuli and/or repeated enactment of a particular behavior. This research aims to provide a comparative study between three cognitive systems/tools designed to allow learning, by using cognitive scripts representation. Since retrieving and re-using past experiences is the core of any learning process, the focus of this thesis is to examine the current existing cognitive systems and tools to evaluate their ability to retrieve past experiences. SOAR, myCBR and Pharaoh are three systems considered for this thesis. Linear and multi-branched cognitive scripts were considered in order to measure the capacity of those systems to allow learning using cognitive scripts representation. The results of this work show that SOAR, myCBR and Pharaoh took almost the same time to retrieve a set of similar cognitive scripts to a query script. However, SOAR was able to retrieve one similar script only, while myCBR and Pharaoh were able to retrieve multiple scripts. Pharaoh tops the other two system in its ability to handle multi-branched scripts of different sizes and the way it considers context.

Table of Contents

ABSTRACT iii
List of Figures: v
List of Tables: v
Chapter 1 Introduction 1
1.1 Motivation 3
1.2 Thesis Goal 3
1.3 Thesis Organization 4
Chapter 2 Background 5
2.1 Literature Review 5
2.2. Cognitive Scripts 6
2.3. Cognitive Systems 10
2.4. Evaluation of cognitive systems 13
Chapter 3 Current Work 14
3.1 SOAR 14
3.2 myCBR 23
3.3 Pharaoh 28
3.5 Comparative Study 31
Chapter 4 Discussion and Conclusion 33
4.1 Discussion and Conclusion 33
4.2 Future Work 36
References 37
Appendix A: Cognitive scripts in SOAR 40
Appendix B: Cognitive scripts in myCBR 45
Appendix C: Cognitive scripts in Pharaoh 46

List of Figures:

Figure 1: Intra-domain scripts 7
Figure 2: Cross-Domain Scripts 9
Figure 3: Cognitive system model for science education system 11
Figure 4: Representation of the script in SOAR 16
Figure 5: Snapshot of SOAR's interface 17
Figure 6: Screenshot of GUI of myCBR. 24
Figure 7: Screenshot of MS Excel Sheet. 24
Figure 8: Dance script is retrieved with 71% similarity 27
Figure 9: Example cognitive scripts by Gawish et al. (Gawish, Abbas, Mostafa, & Salem, 2013) 29
Figure 10: Evolved Multibranched Script TS2 (Gawish, Abbas, Mostafa, & Salem, 2013) 30

List of Tables:

Table 1: Comparison between SOAR, myCBR and Pharaoh Systems 31
Table 2: Time Complexity 31
Table 3: Cognitive tool Strengths and limitations. 32

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Rania Hodhod, and the committee members Dr. Alfredo Perez, Dr. Shamim Khan and Dr. Aisha Patrice Adams for their guidance, and continuous support. I would like to extend my thanks to my parents and family for their encouragement.

Chapter 1 Introduction

The most ambitious goal of computer science is to give the machine the capacity to learn, adapt, organize or repair itself. In the recent years, in complex environment like the Internet, intelligent agents do provide help. Intelligent agents are autonomous entities which have the capacity to learn or use their knowledge to achieve their goals (Carmel & Markovitch, 1996). They are likely to encounter other agents and may need to interact with them to achieve their goals. For example, to obtain highly relevant information at low cost there should be interaction between the information gathering agent and information supplying agent (Carmel & Markovitch, 1996).

When considering efficient strategy for interaction, an agent must consider two main outcomes of its behavior. First, when one agent encounters other agents there should be a direct reward for its action. Second, there should be an effect on the future behavior of the agents by the present behavior (Carmel & Markovitch, 1996). It is a difficult thing to design an effective strategy for interaction because its effectiveness depends mostly on the strategies of other agents involved; each agent has its own private strategies for being autonomous. One solution to this problem is to establish each agent with the ability to adapt their strategies based on their interaction experience. Efficient strategies are required by the agents that operate in a multi-agent system to handle their encounters with other agents involved. Learning capabilities are important for an agent that interacts with another agent (Carmel & Markovitch, 1996).

Intelligent agents have been characterized using mentalistic notions, such as belief, knowledge, intention and commitments (Braziera, Dunin-Kepliczb, Treura, & Verbrugge., 1996). They are provided with learning mechanisms that allows them to acquire new knowledge, and skills which

might involve synthesizing different types of information. There are different techniques which have been used for introducing learning mechanisms in intelligent agents, such as case based reasoning (Mantaras, 2002; Aamodt & Plaza, 1994), analogy (Salem & Gawish, 2016), and conceptual blending (Hodhod, Magerko, & Gawish, 2013).

Cognitive scripts can be seen as a way to enrich intelligent agents' experiences and help them adapt to new situations or work in new environments. Cognitive scripts are a form of memory structures that evolve over multiple exposures to same set of stimuli. In a cognitive script, each event is either temporally or causally linked with the preceding and succeeding events (Ratan, Iyer, & JAMS, 1988). Although different web ontology systems and tools, like aglet, facile have been used to allow learning for intelligent agents, it is not clear how those systems can handle contextual based information in cognitive scripts (Hodhod, Magerko, & Gawish, 2013). To allow learning from existing space of cognitive scripts, a retrieval mechanism is needed to retrieve the right set of previous experiences to be used by the agent. In the retrieval process of cognitive scripts contextual information plays a significant role in identifying the right scripts from large search space of scripts. Accordingly, the application of the standard information retrieval methods might not be ideal because of the type of contextual information in cognitive scripts (Hodhod, Magerko, & Gawish, 2013).

Retrieval of past experiences in humans happens through memory recall, which refers to the subsequent re-accessing of events or information from the past which have been previously encoded and stored in the brain (Newell & H.A.Simon, 1972). During recall, the brain replays a pattern of neural activity that was originally generated in response to a particular event, echoing the brain's perception of the real event. There are two main methods of retrieving memory, they are recognition and recall. Recognition is the association of event or physical object with one

previously experienced or encountered. For example, true or false or multiple choice questions. Recall involves remembering a fact, event or object which is not present currently (physically) and requires the information to be uncovered from the memory. For example, fill-in the blank questions (Newell & H.A.Simon, 1972). Like humans, intelligent agents would also need to retrieve relevant information or past experiences to learn from them or to help the agents deal with a new situation.

As cognitive scripts can successfully represent contextual information like social experiences, this work will explore the ability of existing tools/systems to allow learning using cognitive scripts representation.

1.1 Motivation

Intelligence can be defined as the ability of the system to calculate, reason, learn from experience, store the experience and use them when needed to solve problem and adapt to new situations (Bandura, 1977). It is important to let intelligent agents adopt human learning models in order to be able to handle new experiences in a way similar to humans'. It has been shown that human store their experiences for events in the form of cognitive scripts and learn by expanding those cognitive scripts (adding new events to those scripts). It is important to look at the existing cognitive systems and tools and examine their capability of learning from cognitive scripts. This work will provide a comparison between the cognitive systems and tools in addition to highlighting their strengths and weaknesses in this aspect.

1.2 Thesis Goal

The main goal of this thesis is to know if these existing cognitive systems/tools (SOAR, myCBR and Pharaoh) can allow learning similar to human beings. Since the first step to learn from past

experience is to be able to retrieve the right set of past experiences, this thesis focus on retrieval of past experience.

1.3 Thesis Organization

This thesis is organized as follows: Chapter 2 contains the literature review about human learning and cognitive systems. Chapter 3 presents the current work, discusses several cognitive systems/tools like SOAR, myCBR and pharaoh, in addition to presenting a comparative analysis between those systems and tools. Finally, Chapter 4 presents the thesis conclusion and future work.

Chapter 2 Background

2.1 Literature Review

Human learning is one of the most important cognitive skills that humans possess (Niedderer & Schecker, 2010). Niedderer and Schecker (2010) have formulated a model to distinguish between thinking and learning in the following ways: (1) Thinking is described as processes in the mind where existing cognitive elements can be used for new contexts (frameworks, knowledge). (2) Learning is described as using cognitive elements to change cognitive processes that results from the cognitive system interacting with external situations.

There exist two forms of learning: implicit and explicit learning (Shanks & John, 2012). Implicit learning occurs without concurrent awareness of what is being learned, whereas explicit learning occurs with concurrent awareness. We assume subconscious learning in implicit learning. We assume that evidence for implicit learning derived from artificial grammar learning (It is a grammar that presents the definitive pattern of unconscious learning, here subjects cannot either report the rules of the grammar or explain their performance even though they clearly learn about input domain). The members of this proposed explicit/implicit learning argued that there is clear demonstration of the subjects' ability to encode new information, without being aware of the information (Shanks & John, 2012).

Generally, humans learn by gaining new experiences or by continuous practice. Learning includes knowledge and intellectual skills, attitudes and emotional responses. There are several theories to explain how learning happens, such as cognitive learning theories (CIPD, 2002), social learning theories (Bandura, 1977), and constructivist theories (Glasersfield, 1989; Bredo, 1994). In cognitive learning theories, learning is viewed as a process of understanding the

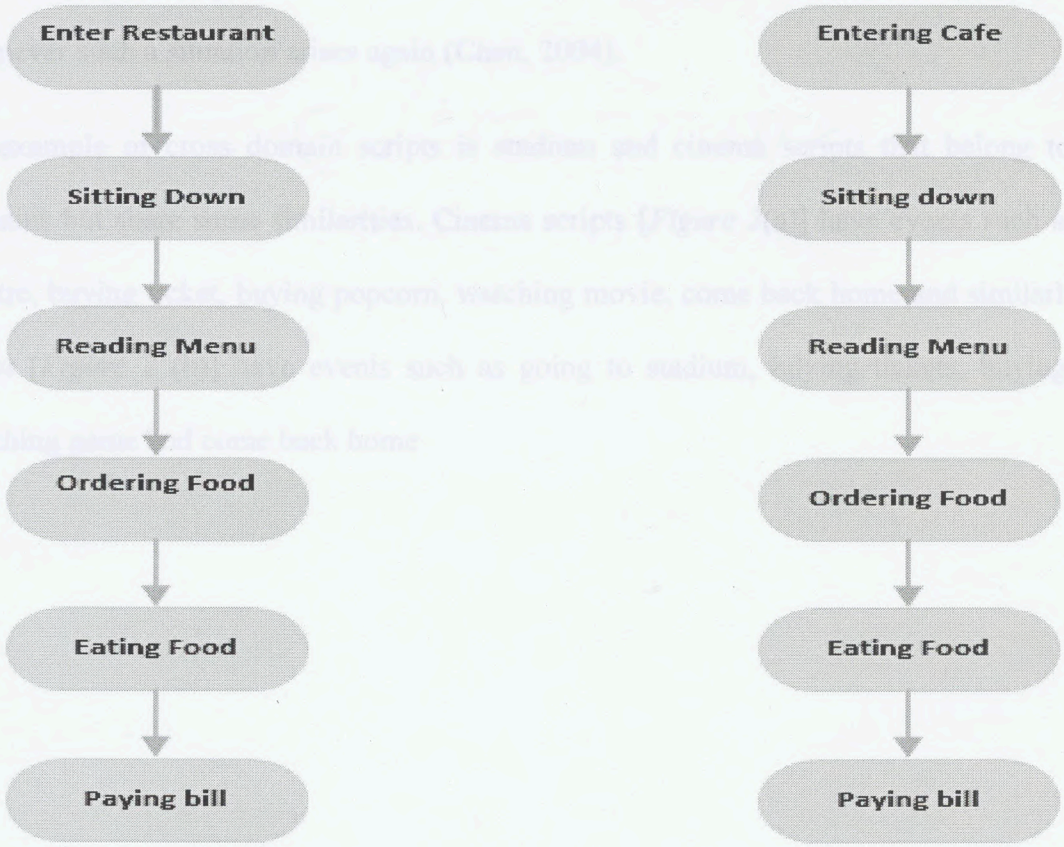
aspects/things of the world around us. Distinct stages of cognitive development are present where processing of information takes place in each stage. Appropriate content (information) is matched to the developmental stage for learning.

Social interaction is required for cognitive development and learning is restricted to a certain range (doing it alone and doing it with help) at a given range (range of thinking capacity). In social learning theories, it is believed that learning happens by observing things. Humans may learn from model behavior hence both humans and environment are reciprocal determinants of each other (Bandura, 1977). In constructivist theories, humans construct cognitive schema (script) by using knowledge they already possess and new information that is present. That cognitive script is stored and reused based on the situation. As per this theory, learning is an active process in which existing knowledge is used to develop new ideas (Mahar & Harford, 2004). Hence cognitive scripts play an important role in the human memory storage.

2.2. Cognitive Scripts

According to Schank and Abelson, "a script is a predetermined, stereotyped sequences of actions that defines a well-known situation" (Schank & Abelson, 1977). Scripts have two properties: (1) scripts are made up of slots and connection between them. Each slot in the script specifies actions in the sequence. (2) connections/links between slots are temporal and causal, these links make the script. In human memory, knowledge (knowledge of events) is represented as a series of actions with a goal to establish (Chen, 2004). For example, a cognitive script for going to a theatre to watch a movie involves series of actions such as going to theatre, buying ticket, buying popcorn/drinks, entering auditorium, watching movie, movie finishes and going home. Humans learn events using cognitive scripts through repetitive social interaction and use them to predict, interpret, and understand new experiences (Nelson., 1986).

Cognitive scripts can allow two types of learning: intra-domain and cross-domain. The first type of learning is intra-domain in which similarities are restricted to provide within the same domain. In intra-domain, surface similarities are confined within the same domain for example restaurant and cafe scripts. There are similar events between these two scripts, such as enter place, sitting down and reading menu, see Figure 1(a) and Figure 1(b).



a. Restaurant script

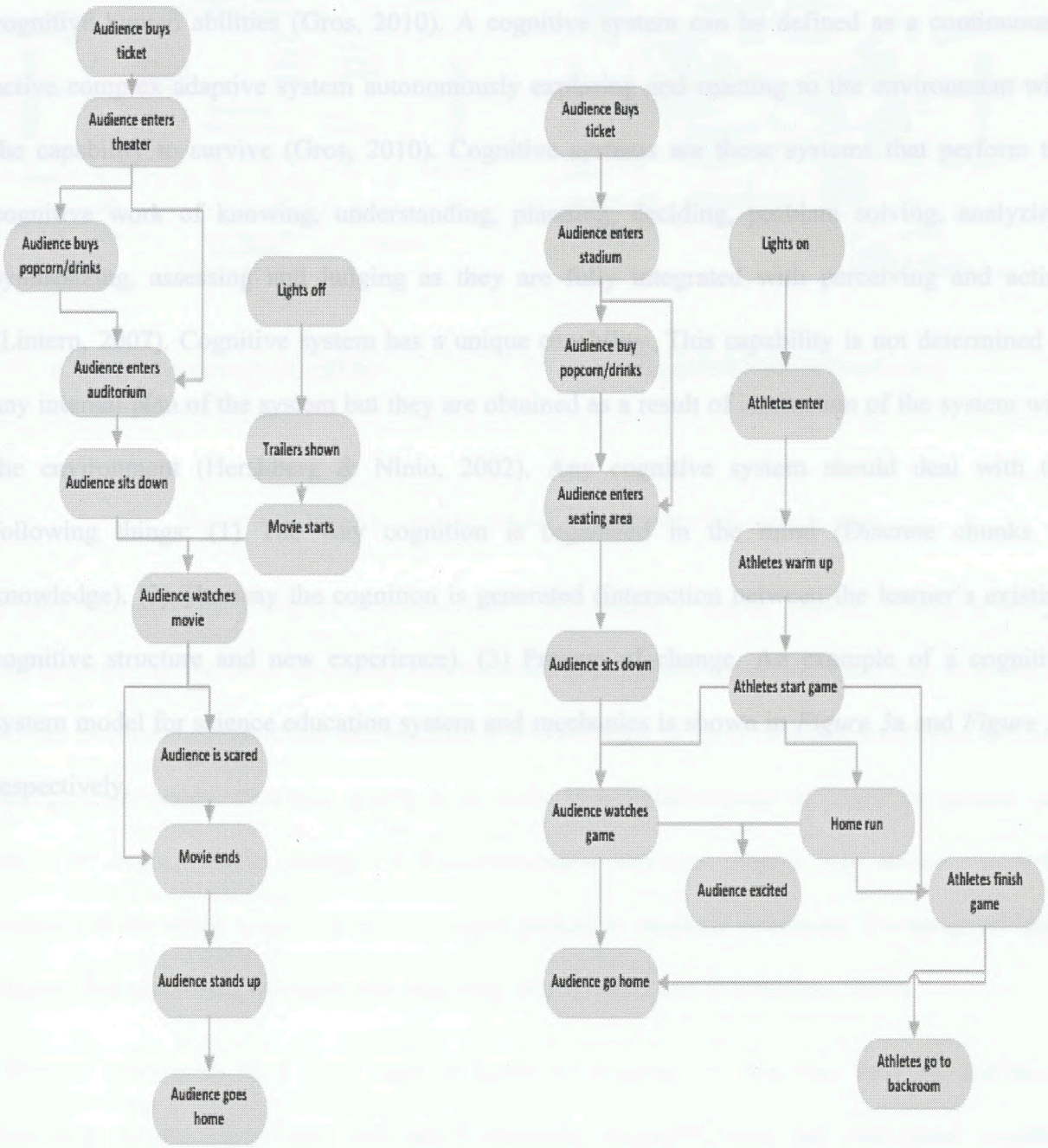
b. Cafe script

Figure 1: Intra-domain scripts

In cross-domain learning, people have an ability to adapt new situations within different domains with the help of deep structural similarities between new and old situations, which is something cognitive scripts highly supports (Salem & Gawish, 2016). Despite the types of learning, human

also learn by experience, experience tells us that a different state of a related object could change the action sequence. For example: a script for going to restaurant can be described in this way; going to restaurant, order food, eat, pay the bill, give tip but sometimes there might be some changes like the food we wish to order might be over or waiter might not respond well then we will leave the restaurant. This situation will be different from other situations scripted before, which creates a problem. This situation will be stored in the memory and reused by humans whenever such a situation arises again (Chen, 2004).

An example of cross domain scripts is stadium and cinema scripts that belong to different domains but share some similarities. Cinema scripts [*Figure 2*(a)] have events such as going to theatre, buying ticket, buying popcorn, watching movie, come back home and similarly Stadium script [*Figure 2* (b)] have events such as going to stadium, buying tickets, buying popcorn, watching game and come back home



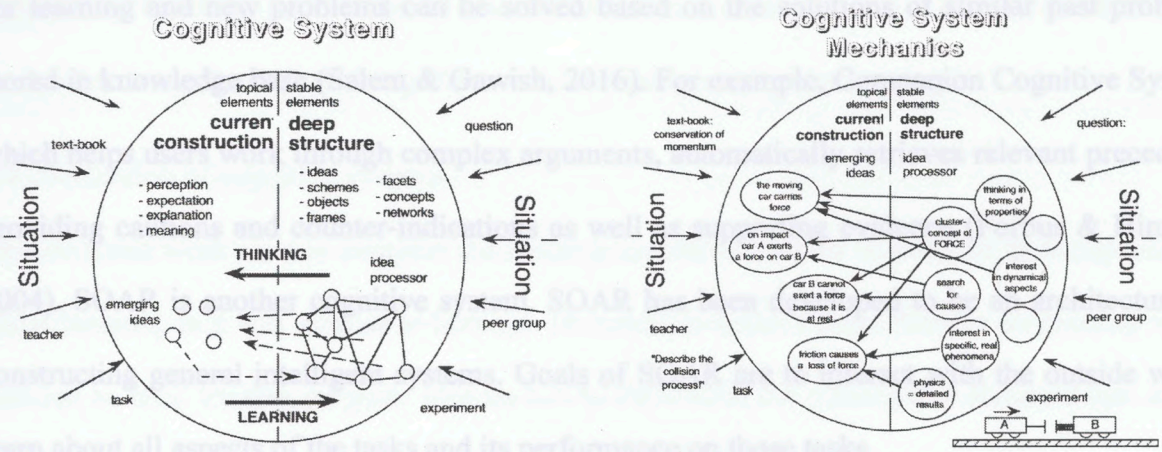
a. Cinema script

b. Stadium Script

Figure 2: Cross-Domain Scripts

2.3. Cognitive Systems

There have been continuous efforts to design and develop cognitive systems that model various cognitive human abilities (Gros, 2010). A cognitive system can be defined as a continuously active complex adaptive system autonomously exploring and reacting to the environment with the capability to survive (Gros, 2010). Cognitive systems are those systems that perform the cognitive work of knowing, understanding, planning, deciding, problem solving, analyzing, synthesizing, assessing and judging as they are fully integrated with perceiving and acting (Lintern, 2007). Cognitive system has a unique capability. This capability is not determined in any internal plan of the system but they are obtained as a result of interaction of the system with the environment (Hershberg & Ninio, 2002). Any cognitive system should deal with the following things: (1) The way cognition is organized in the mind (Discrete chunks of knowledge). (2) The way the cognition is generated (interaction between the learner's existing cognitive structure and new experience). (3) Process of change. An example of a cognitive system model for science education system and mechanics is shown in *Figure 3*a and *Figure 3*b respectively.



a. Cognitive system model for science education system

b. Cognitive system Mechanics

Figure 3: Cognitive system model for science education system

Figure 3(a) presents a cognitive system model with ideas, schemas, objects, frames, concepts and networks as the deep structures and perception, expectations, explanation and meaning as current construction. Thinking and learning is the exchange of ideas between deep structures and current construction. Learning seems to be seen as a stable change in cognitive system that allows to explain stable changes in the individual's behavior. *Figure 3(b)* shows a specific example of the above cognitive system model applied to students' reasoning. Exchange of ideas happens between idea processor and emerging ideas (Niedderer & Schecker, 2010).

Different techniques have been used in cognitive systems for learning, such as analogical reasoning, genetic algorithms, case based reasoning, linguistic rules, and conceptual blending (Carmel & Markovitch, 1996). Analogical reasoning can be defined as a cognitive process of transferring information from the known domain to the less familiar domain based on similarity between these domains (Gentner & Smith, 2012). Cognitive systems used analogical reasoning

for learning and new problems can be solved based on the solutions of similar past problems stored in knowledge base (Salem & Gawish, 2016). For example, Companion Cognitive System, which helps users work through complex arguments, automatically retrieves relevant precedents, providing cautions and counter-indications as well as supporting evidence (Forbus & Hinrichs, 2004). SOAR is another cognitive system. SOAR has been developed to be an architecture for constructing general intelligent systems. Goals of SOAR are to interact with the outside world, learn about all aspects of the tasks and its performance on those tasks.

Case-based reasoning (CBR) explicitly integrates memory, learning, and reasoning. The case processor will be present in CBR which has variety of responsibilities. It helps to carry out processing and finding appropriate cases in memory, applying them in a new situation, and learning. Steps involved in case processor are:

- New situation is interpreted in such a way that cases that are relevant can be located in the case library.
- Deciding which old case is most applicable.
- Lessons learned from old cases are applied to the new situation.
- Experience is structured as a case and choosing ways of indexing it when necessary.
- Re-interpreting and re-indexing an old case in the presence of new findings.

These steps are important for productive use of cases for reasoning and learning. Examples of cognitive CBR systems for learning are Reflective Learner and Archie (Kolodner & Guzdial, 2000). The Reflective Learner provides prompts, informed by CBR, to help students write learning essays in a more effective manner. Archie is a case based design aid for professional architects. It intends to provide consultations whenever designers, who are working on the design

of a public building, need some advice. STABLE is a descendent of ARCHIE-2 which helps students to learn the skill of doing object oriented design and programming.

2.4. Evaluation of cognitive systems

This research work aims to compare the existing cognitive systems to evaluate their capability to allow learning in a way similar to humans. However, it doesn't seem there is an existing metric that can be used for that purpose. Metrics can be defined as the prescriptive standard that is used to determine the superiority of a system over another (Langley & Messina, 2005) and can be seen as a form of frames where attributes and values are present. Metric evaluation using frames might be useful and easy to use (Chen, 2004).

Metrics can be used to compare and improve any model as metrics measures the quality of model predictions (Albino, Garavelli, & Schiuma, 2000). They are used for parameter fitting, since many fitting procedures aim to optimize parameters with respect to some metric (Pelanek, 2015). In a research process, the choice of metric is an important step. Metrics can be in terms of not accurately, almost accurately and accurately measurable. In these cases, the inputs of the process are mainly known, and the possibility of defining their characteristics depends on the metrics which can be adopted for their evaluation (Albino, Garavelli, & Schiuma, 2000); if metrics are qualitative and/or subjective they are expressed by adjective scale (what is adjective scale), this kind of metric needs expert's evaluation. Here inputs are defined as non-accurately measurable. If metrics to evaluate the input characteristics are mainly quantitative and/or objective they are expressed by numerical scales (example of numerical scale), inputs are defined as almost accurately measurable. If the input set can be scientifically described (what is scientifically described), then they can be defined as accurately measurable. This is evaluated by quantitative and objective metrics.

Chapter 3 Current Work

Cognitive systems for learning have shown success when dealing with knowledge representation like rules and frames. It is not clear how these cognitive systems/tools would behave when presented with social experiences represented in a form similar to that experienced by humans, that is cognitive scripts. The existing cognitive systems/tools like SOAR, myCBR, and Pharaoh have been chosen for this thesis to perform comparative study between these systems/tools for event based retrieval.

3.1 SOAR

SOAR is a computational model for knowledge representation and manipulation (Schalkoff, 2011) Practically, SOAR is based upon the notion of a production system. The major goal of the SOAR project is to fully implement the capabilities of an intelligent agent. The knowledge in the SOAR agent is represented as if- then rules, known as productions. SOAR works by testing the "if" parts of the rule, if all the conditions are true, "then" parts are executed; all activities in SOAR takes place in the working memory. SOAR has three types of memories; working memory (episodic memory), production memory and preference memory. Working memory consists of state, this memory is called as short-term memory, production memory consists of user-defined and "learned" productions, preference memory facilitates conflict resolution, and episodic memory is a memory of specific events in our past which uses past experiences to anticipate future events.

SOAR has been used successfully in many projects. One project that used SOAR is Viewpoints AI (Jacob & Magerko, 2015). ViewPoints AI system features procedural gesture interpretation using shallow semantics and deep aesthetics analysis from the Viewpoints framework. The installation creates a liminal virtual / real space for the human and AI to interact by the use of

digital projection for the AI visualization and shadow play to represent the human. The system uses case-based reasoning (CBR) to index and store the agent's experiences in the form of cases that can be utilized to drive future behavior or responses in general. SOAR matches past experiences from its episodic memory and choose a similar gesture to bring into the new interaction context. This is achieved by SOAR's episodic memory partial graph matching capabilities. SOAR has a set of strategies that are used for selecting responses to the input gesture which are then output to the action module to produce the reaction gesture.

Another project that used SOAR is Rosie (Kirk, Mininger, & Laird, 2016). Rosie is a robotic agent that uses a Microsoft Kinect sensor and robotic arm affixed to a tabletop, and is controlled by procedural, declarative, and episodic knowledge encoded in SOAR. Blocks of various sizes, shapes and color can be detected and manipulated by the agent. The parser implemented in SOAR is integrated with Rosie to take a grammatical English sentence as input and produce a semantic interpretation (includes grounding all reference to objects and locations).

Another project that used SOAR is the interactive task learning mobile robot (Mininger & Laird, 2016). In this project, the agent carries out simple set of actions such as movement actions (drive-to-location, turn), manipulation actions (pick-up, put-down), and communicative actions (ask, say). Knowledge of the actions of agent is encoded as rules in SOAR's procedural memory. Agent knows when to perform an action, how to execute the action, and the effects of the action.

As seen from the literature review, none of the above systems used cognitive scripts as way to represent knowledge in SOAR. For the purpose of this thesis, seven cognitive scripts were stored in SOAR's episodic memory to examine how SOAR can handle social scripts represented in the form of cognitive scripts. *Figure 4* shows an example code of a cognitive script representation in SOAR. And *Figure 5* shows a snapshot for SOAR's interface.

```
sp {apply*record-story*I
(state <s> ^operator <op>
^next-story I)
(<op> ^name record-story) -->
(<s> ^next-story 2 I -
^event |Customer arrives at a car dealer |
^next <e2>^next <e3>)
(<e2> ^event |Customer looks at car |
^next <e4>)
(<e3> ^event |Agent sees customer |
^next <e4>)
(<e4> ^event |Agent talks to customer |
^next <e5>)
(<e5> ^event |Agent learns about customer |
^next <e6>)
(<e6> ^event |Agent recommends a car |
^next <e7>
^next <e8>)
(<e7> ^event |Customer decides to buy the car |
^next <e9>)
(<e8> ^event |Customer decides to buy a different car |
^next <e9>)
(<e9> ^event |Agent tells customer price |
^next <e10>
^next <e11>)
(<e10> ^event |Customer accepts price |
^next <e12>
^next <I4>)
(<e11> ^event |Customer negotiates price |
^next <e13>)
(<e13> ^event |Customer and agent decide price |
^next <e12>
^next <e14>)
(<e12> ^event |Customer pays in cash|
^next <e16>)
(<e14> ^event |Customer get a car loan|
^next <e15>)
(<e15> ^event |customer and agent sign files |
^next <e16>)
(<e16> ^event |Customer drives car home|)}
```

Figure 4: Representation of the script in SOAR

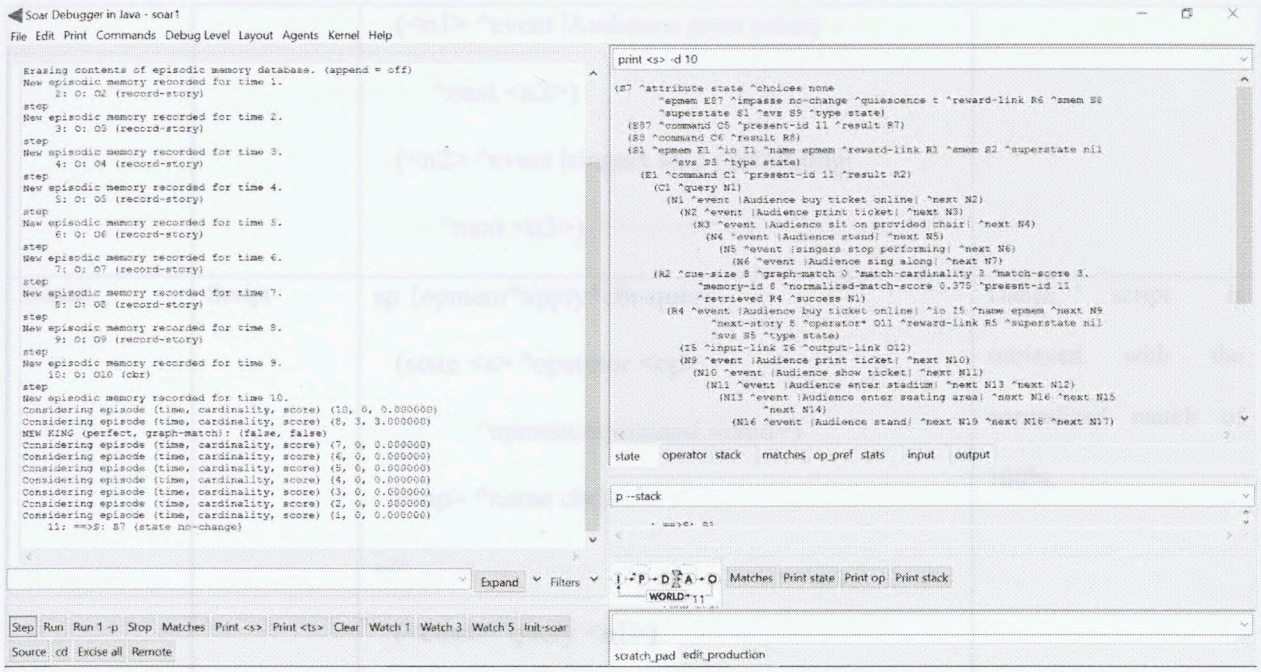


Figure 5: Snapshot of SOAR's interface

Seven cognitive scripts were stored in the episodic memory of the SOAR (Car-buying, Cinema, Classical, Dance, Pharmacy, Restaurant, Stadium). SOAR was then queried by a set of query scripts ranging from a subscript of one of the existing scripts in SOAR's memory to a completely new script. Results from these queries are presented in the table below:

Script Name	**Description**	**Code**	**Results**
Dance	Subscript from Dance script	sp {epmem*apply*cbr-query (state <s> ^operator <op> ^epmem.command <cmd>) (<op> ^name cbr) --> (<cmd> ^query <n1>)	Dance script is retrieved with the normalized match of 33%.

		(<n1> ^event |Audience print ticket| ^next <n2>) (<n2> ^event |singers stop performing| ^next <n3>)	
Dance	Script	sp {epmem*apply*cbr-query (state <s> ^operator <op> ^epmem.command <cmd>) (<op> ^name cbr) --> (<cmd> ^query <n1>) (<n1> ^event |Audience buy ticket online| ^next <n2>) (<n2> ^event |Audience print ticket| ^next <n3>) (<n3> ^event |Audience show ticket| ^next <n4>) (<n4> ^event |Audience enter stadium| ^next <n5> ^next <n6>) (<n5> ^event |Audience buy snacks drinks| ^next <n6>) (<n6>^event |Audience enter seating area| ^next <n7> ^next <n8> ^next <n9>)	Dance script is retrieved with the normalized match of 100%.

		(<n7> ^event |Audience sit on own chair| ^next <n10> ^next <n11>) (<n8> ^event |Audience sit on provided chair| ^next <n10> ^next <n11>) (<n9> ^event |Audience stand| ^next <n10> ^next <n11> ^next <n12>) (<n10> ^event |Audience listen to song| ^next <n13>) (<n11> ^event |Audience sing along| ^next <n13>) (<n12> ^event |Audience dance| ^next <n13>) (<n13> ^event |singers stop performing| ^next <n14> ^next <n15>) (<n14> ^event |singers go to backroom|) (<n15> ^event |audience go home|) }	
Dance	Longer subscript from	sp {epmem*apply*cbr-query (state <s> ^operator <op> ^epmem.command <cmd>)	Dance script is retrieved with the normalized match

	Dance script	(<op> ^name cbr) --> (<cmd> ^query <n1>) (<n1> ^event |Audience buy ticket online| ^next <n2>) (<n2> ^event |Audience print ticket| ^next <n3>) (<n3> ^event |Audience sit on provided chair| ^next <n4>) (<n4> ^event |Audience stand| ^next <n5>) (<n5> ^event |singers stop performing| ^next <n6>) (<n6> ^event |Audience sing along| ^next <n7>) (<n7> ^event |Audience dance| ^next <n8>) (<n8> ^event |singers go to backroom|) }	of 37.5%.
Hotel	New script	sp {epmem*apply*cbr-query (state <s> ^operator <op> ^epmem.command <cmd>) (<op> ^name cbr) --> (<cmd> ^query <n1>)	No script is retrieved.

		(<n1> ^event |People enter hotel| ^next <n2>) (<n2> ^event |People order food| ^next <n3>) (<n3> ^event |People eat ordered food| ^next <n4>) }	
Dance	Three Linear events	sp {epmem*apply*cbr-query (state <s> ^operator <op> ^epmem.command <cmd>) (<op> ^name cbr) --> (<cmd> ^query <n1>) (<n1> ^event |Audience buys snacks drinks| ^next <n2>) (<n2> ^event |Audience enter seating area| ^next <n3>) (<n3> ^event |Audience_sit_on_provided_chair| ^next <n4>) }	Dance script is retrieved with the normalized match score of 25%.

Dance	Three Non-Linear events	sp {epmem*apply*cbr-query (state <s> ^operator <op> ^epmem.command <cmd>) (<op> ^name cbr) --> (<cmd> ^query <n1>) (<n1> ^event |Audience_print_ticket| ^next <n2>) (<n2> ^event |Audience_enter_stadium| ^next <n3>) (<n3> ^event |Audience_sit_on_provided_chair| ^next <n4>) }	Dance script is retrieved with the normalized match score of 25%.

Subscripts from Dance and hotel scripts were considered to query SOAR. Firstly, a subscript from the Dance script was used as a query script. The Dance script was retrieved with the normalized match of 33%. Secondly, the whole script of the Dance script was considered, Dance script is retrieved with the normalized match of 100%. Thirdly, a subscript considered is longer subscript (whole branch of Dance script) and Dance script is retrieved with the normalized match of 37.5%. New script is taken as a subscript, as there is no matching with stored seven scripts, the normalized match obtained is zero percentage with no script retrieved. Three linear and non-linear subscripts from the Dance script were used for retrieval; every time the Dance script was

retrieved with a 25% normalized match score. These matching percentages are obtained based on the length of the subscript and percentage of matching with the stored script.

3.2 myCBR

myCBR is an open-source similarity-based retrieval tool which can test highly sophisticated, knowledge-intensive similarity measures in a powerful GUI (Roth-Berghofer & Garcia, 2011)(see *Figure 6*). myCBR has been used successfully in many projects. One project that used myCBR is Creating knowledge from Unstructured Documents. This approach has been developed in machine diagnosis based on experimental knowledge from engineers. (Bach, Althoff, Newo, & Stahl, 2011). Throughout the project they kept using the myCBR workbench when refining case formats as well as similarity measure.

One project that used myCBR is Knowledge Formalisation for Audio Engineering. (Sauer, Roth-Berghofer, Auricchio, & Proctor, 2013). The approach to formalize the special vocabulary used in audio engineering, consisting of vague descriptors for timbres, amounts and directions was developed. CBR was introduced as a methodology to amend the problem of formalizing the vagueness of terms. The myCBR 3 Workbench is used to swiftly transfer their initial elicited knowledge model into a structured CBR knowledge model.

Another project that used myCBR is Knowledge Formalisation for Hydrometallurgy Gold Ore Processing. (Sauer, Rintala, & Roth-Berghofer., 2013). In this case study research is performed to elicit and formalize knowledge in the domain of hydrometallurgical processing of gold ore.

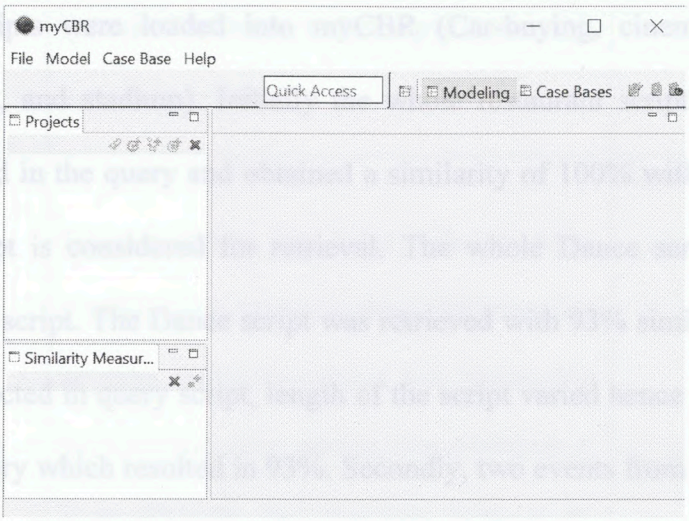


Figure 6: Screenshot of GUI of myCBR.

For the purpose of this thesis, seven cognitive scripts were loaded into myCBR using a .csv document, see *Figure 7*.

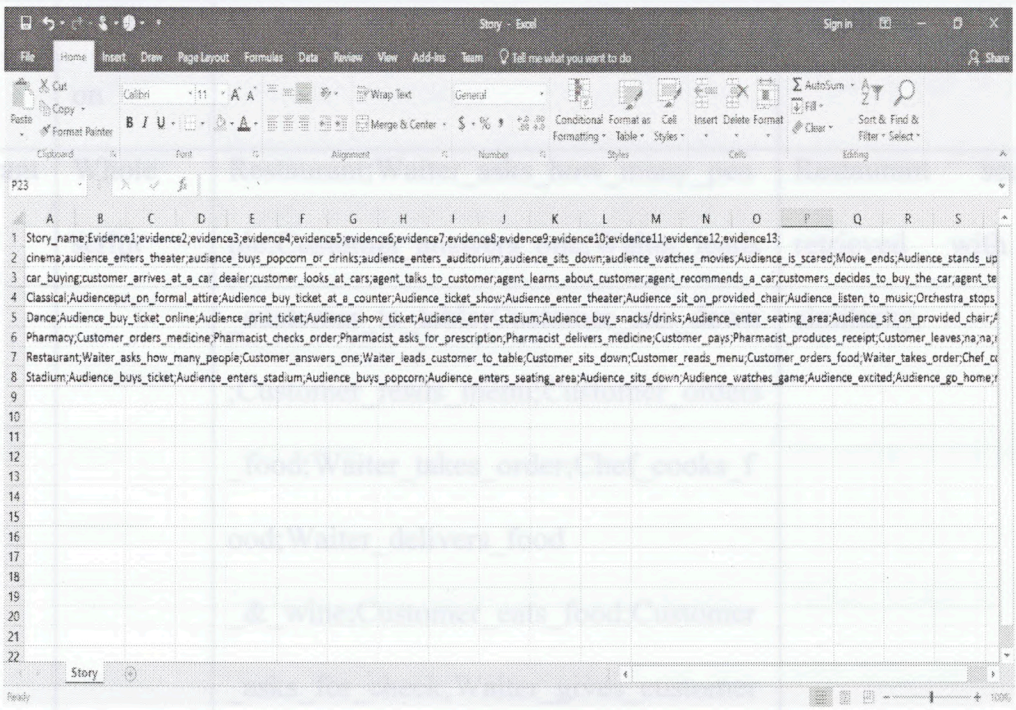


Figure 7: Screenshot of MS Excel Sheet.

Seven cognitive scripts were loaded into myCBR (Car-buying, cinema, classical, concert, pharmacy, restaurant, and stadium). Initially the whole restaurant script was considered as a subscript and selected in the query and obtained a similarity of 100% with the restaurant script. Then Dance subscript is considered for retrieval. The whole Dance script is considered and selected as the query script. The Dance script was retrieved with 93% similarity even though the whole script was selected in query script, length of the script varied hence the "unknown" has to be selected in the query which resulted in 93%. Secondly, two events from the Dance script were used in the query, as a result the Dance script was retrieved with 38% similarity. Thirdly, a longer subscript (whole branch) from the Dance script was used as the query script, the obtained result was retrieving the Dance with 71% similarity measure. Three linear and non-linear events from the Dance script were used as the query script and obtained 46% similarity measure.

Results from testing myCBR are shown below:

Script Name	Description	Code	Results
Restaurant	Whole script	Restaurant;Waiter_asks_how_many_people;Customer_answers_one;Waiter_leads_customer_to_table;Customer_sits_down;Customer_reads_menu;Customer_orders_food;Waiter_takes_order;Chef_cooks_food;Waiter_delivers_food_&_wine;Customer_eats_food;Customer_asks_for_check;Waiter_gives_customer	Restaurant script is retrieved with 100% similarity.

		_check;Customer_writes_tip;Customer_leaves	
Dance	Script with na	Dance;Audience_buy_ticket_online;Audience_print_ticket;Audience_show_ticket;Audience_enter_stadium;Audience_buy_snacks/drinks;Audience_enter_seating_area;Audience_sit_on_provided_chair;Audience_listen_to_song;Audience_dance;Singer_stops_performing;Audience_go_home	Dance script is retrieved with similarity of 93%
Dance	Subscript from Dance script	Dance;Audience_print_ticket;Singer_stops_performing	Dance script is retrieved with the similarity of 38%.
Dance	Longer subscript form Restaurant script	Dance;Audience_buy_ticket_online;Audience_print_ticket;Audience_sit_on_provided_chair;Audience_listen_to_song;Audience_dance;Singer_stops_performing	Dance script is retrieved with the similarity of 71%
Unknown	New script	_unknown_	No script is retrieved.
Dance	Three linear events	Audience_buy_snacks/drinks;Audience_enter_seating_area;Audience_sit_on_provided_chair	Dance script is retrieved with 46% similarity

Dance	Three non-linear events	Audience_print_ticket;Audience_enter_stadium; Audience_sit_on_provided_chair	Dance script is retrieved with 46% similarity

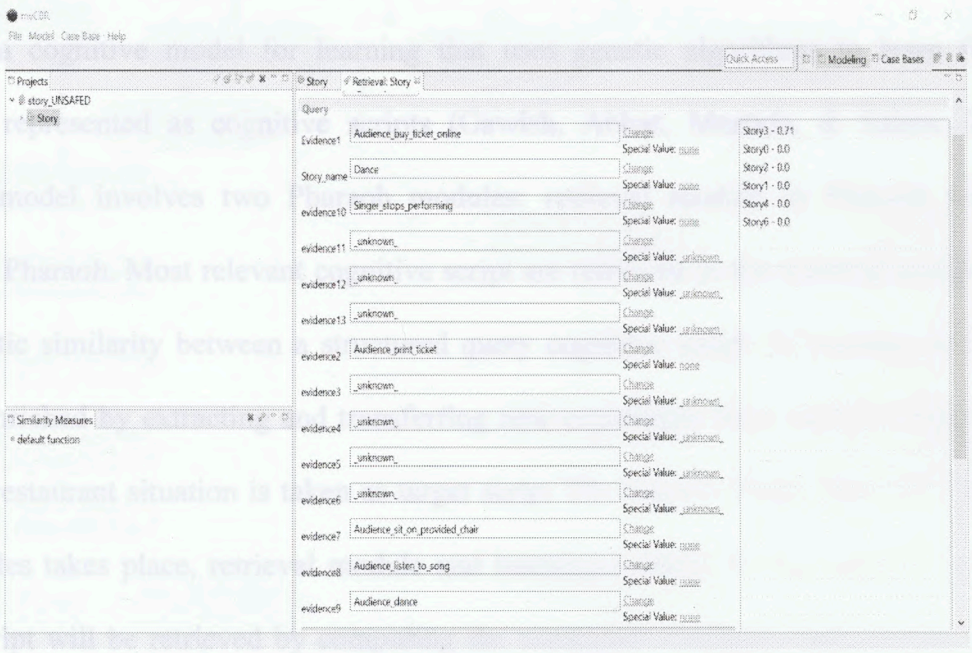


Figure 8: Dance script is retrieved with 71% similarity

In *Figure 8* few events of dance script are selected in query and similarity obtained is 71% with story 3 (i.e. Dance script).

Similarity is obtained based on the percentage of matching between the query script and stored script. Exact word to word matching is considered in myCBR. Similarity measure between the events in a query script and events in pre-stored script provides the similarity percentage. In *Figure 8* how events from the Dance subscript were entered in myCBR is shown. During the retrieval process, the similarly measure is obtained based on the degree of matching between the

query subscript and stored scripts. In *Figure 8* the resulting similarity is 71% as the events in the query subscript partially matches the pre-stored Dance script (query events define a subset of the actual Dance script).

3.3 Pharaoh

Pharaoh is another cognitive system that is used to retrieve the complete script. Gawish et al. has proposed a cognitive model for learning that uses genetic algorithms to learn from social situations represented as cognitive scripts (Gawish, Abbas, Mostafa, & Salem, 2013). The proposed model involves two Pharaoh modules: retrieval module in Pharaoh and learning module in Pharaoh. Most relevant cognitive script are retrieved in the retrieval module based on the semantic similarity between a structured query cognitive script. In learning module, target script is enriched by extracting and transferring new experience from retrieved base script. For example, restaurant situation is taken as target script TS_1 (*Figure 9* (a)). Here above mentioned two modules takes place, retrieval module and learning module. In the retrieval module, most similar script will be retrieved by computing the contextual similarity between the TS_1 and all other scripts. Retrieval phase returns pharmacy script (*Figure 9*(b)) as most similar script to TS_1. In learning module TS_1 will be improved in order to yield an evolved version TS_2 (*Figure 10*) where some related events from pharmacy script are added to the restaurant script. (Gawish, Abbas, Mostafa, & Salem, 2013).

a. Multibranched restaurant script TS_1 b. Multibranched pharmacy script

Figure 9: Example cognitive scripts by Gawish et al. *(Gawish, Abbas, Mostafa, & Salem, 2013)*

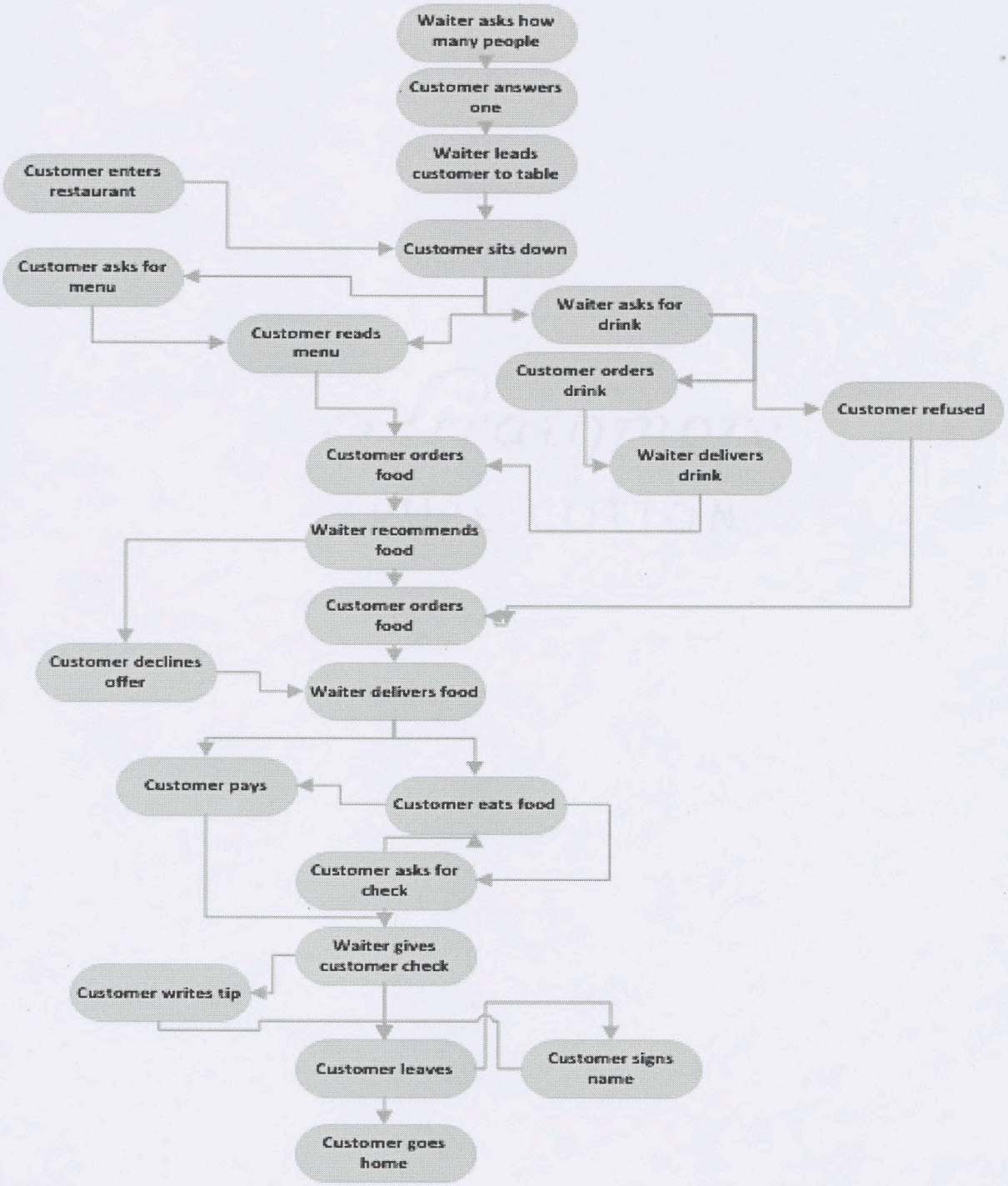


Figure 10: Evolved Multibranched Script TS2 *(Gawish, Abbas, Mostafa, & Salem, 2013)*

3.5 Comparative Study

Seven scripts (Car-buying, Cinema, Classical, Dance, Pharmacy, Restaurant and Stadium) were stored in all three systems cognitive systems; SOAR, myCBR and Pharaoh. *Table 1* shows how each of the systems performed when introduced with cognitive scripts structures. *Table 3* describes the strengths and limitation of SOAR, myCBR and Pharaoh when using cognitive scripts.

Table 1: Comparison between SOAR, myCBR and Pharaoh Systems

Query Script	**SOAR Most similar scripts in order**			**myCBR Most similar scripts in order**			**Pharaoh Most similar scripts in order**		
	1	2	3	1	2	3	1	2	3
Car-buying	Car-buying	None	None	Car-buying	None	None	Pharmacy	None	None
Cinema	Cinema	None	None	Cinema	Stadium	Dance	Stadium	Dance	Classical
Classical	Classical	None	None	Classical	Dance	None	Dance	Cinema	Stadium
Dance	Dance	None	None	Dance	Stadium	Classical	Classical	Cinema	Stadium
Pharmacy	Pharmacy	None	None	Pharmacy	None	None	Restaurant	Car-Buying	None
Restaurant	Restaurant	None	None	Restaurant	None	None	Pharmacy	None	None
Stadium	Stadium	None	None	Stadium	Cinema	Dance	Cinema	Concert	Classical

Seven scripts Car-buying, Cinema, Classical, Dance, Pharmacy, Restaurant and Stadium) were loaded into all the three systems (SOAR, myCBR, Pharoah) and most similar script is being retrieved. None is indicated if no similar script is present. SOAR was able to retrieve same script but it could not retrieve more than one script whereas myCBR could retrieve more than one script but not based on the context. Pharaoh was only one which could retrieve more than one script based on context.

Table 2: Time Complexity

Tools	Average time taken for execution	
	Linear scripts	Multibranched scripts
SOAR	00:00:02.496	Not applicable
myCBR	00:00:00.813	00:00:00.813
Pharaoh	00:00:00.2	00:00:515 to 00:00:10

Table 2 shows the time complexity of SOAR, myCBR and Pharaoh. Pharaoh has the fastest average time to retrieve linear scripts.

Table 3: Cognitive tool Strengths and limitations.

Tools	Strengths	Limits
SOAR	-Code is represented in the form of if then rules. -Exact similarity measure is obtained. -Multiple branches can be represented	-Single script is retrieved -Even though it can represent multiple branches it should have single root.
myCBR	-Simple English is used and data is loaded by saving the data into excel sheet.	-Length of the scripts should be same if not then 'na' will be added at the end which results in invalidate similarity measure. -Cannot represent multiple branches
Pharaoh	-Can represent multiple branches -It can have multiple roots	-does not contain internal mechanism to evaluate the output blended script.

SOAR, myCBR and Pharaoh have their own strengths and limitations but all the three systems/tools were very efficient in retrieval of scripts. For example, SOAR retrieves exactly matching script but it can retrieve only one script at a time. myCBR is simple, easy to use but whereas SOAR was more challenging to use as it involves complex codes. Pharaoh can have multiple roots (multiple starting points), this feature is not present in any of the systems/tools (SOAR, myCBR). Each of the systems/tools have great strengths like multiple branch representation (SOAR and Pharaoh), representation of query in simple English (myCBR).

Chapter 4 Discussion and Conclusion

4.1 Discussion and Conclusion

The human beings, in their thinking and reasoning, they apply a schema in an attempt to use it alter in the process of reconstruction of an organized thought. With the help of this schema, the thoughts and the individual behavior are tuned in a way that it helps the individuals to develop and subconsciously maintain some relationships with the rest of the thoughts they have (Bandura, 1977).

A script can be considered to be a stereotypical and predetermined set of action and sequences that are used in the process of defining a situation that is ideally well-known. In this consideration, the scripts are considerably known to have the properties of a stable connection between them as well as the fact that they are essentially made of slots between each of the connections. Linking the cognitive scripts to the aspects of human memory, one thing that is prevalent is based on the prepositive that the knowledge attributes of the humans are manifested in the form of a series of actions. In this case, these actions are stored in the brain with an inherent pursuit and an awaiting completion goal.

In an attempt to allow the virtues of learning from experiences represented as cognitive scripts, there is substantially a need for a mechanism to retrieve the appropriate experiences for them to be applied and used as an agent. In this case, it is fundamentally important to keep in mind that this is bound by the type of contextual information that is found to be embedded in cognitive scripts. Schemas such as cognitive scripts and models are seen to be used in the process of explaining the repeated enactment of a particular behavior.

In humans, the application of the cognitive scripts is fundamentally important as it helps them in the process of learning. They also help individuals to interpret, understand and be able to predict the experiences. The cognitive scripts are also known to allow distinct types of learning among the individuals. These pertain to the cross-domain as well as the intra-domain. These domains are important by the way that they help in the process of assisting the individuals to adapt to new situations.

The computer science field has evolved to be a form of technology that is being embedded in the machines with the capacity to learn, adapt and repair themselves. The computer science models are being capacitated with agents that are needed for their interaction in order to achieve some goals. In the process of making a consideration as to the efficient strategy for interaction, an agent has the sole responsibility of ensuring that they have duly considered the outcome of their behaviors. It has been identified that learning capabilities are important for an agent that interacts with another agent (Carmel & Markovitch, 1996).

Intelligent agents are considerably seen to be using the mentalistic notions which include the values of commitment and intentions in their applicability. There exist explicitly different techniques and applications that have been put in place so as to ensure that these intelligence agents work to suit their design applications and the primary purpose.

When it comes to the cognitive systems, these are characterized as the systems that help the people in the process of deciding, understanding different phenomena, assessing, planning as well as judging different aspects in their lives. There are different analytical methods that have been applied to try and explain the application of the cognitive systems in the learning process. Both the past and the present problems can be accurately dealt with through the cognitive systems.

Examples of the cognitive systems include the case-based reasoning (CBR) which is predominantly important in the process of assisting in finding out the appropriate cases in the memory. In order for this cognitive system to effectively be in full operation, it requires that there is an application of metrics. Another important cognitive system that was scrutinized relates to the SOAR. In this model, it explicitly aims at implementing the inherent capabilities of any intelligent agent. The memories that this system contains makes use of individual past experiences in order to make an accurate anticipation of all the future events. This system has been accurately applied in several projects which amongst them include Rosie.

Another important cognitive tool that is increasingly being applied in the search for harnessing the past experiences relates to myCBR. This model is applied in various projects that seek to test the implication of the sophisticated and the knowledge-intensive measures. As the research revealed, this cognitive tool is, however, important in its application as it helps in the process of arranging and classifying the thoughts and the past experiences of the individuals.

However, in each of these cognitive systems and tools, there are strengths and weaknesses that they have, especially when it comes to dealing with experiences resented as cognitive scripts.

The cognitive systems like SOAR and Pharaoh were able to handle multiple branched scripts as query scrips and retrieve similar multi-branched scripts. Whereas myCBR could not use multi-branched scripts, this is the main drawback in myCBR. myCBR requires the length of the scripts to be same, else obtained similarity will not be correct. On the other hand, SOAR and Pharaoh support using scripts of different length. For query script, SOAR was able to retrieve one similar script only, while myCBR and Pharaoh allow the retrieval of multiple scripts. To conclude, Pharaoh tops the other systems in its ability to handle multi-branched scripts of different sizes. And its ability to retrieve most relevant/similar scripts considering the context of scripts.

4.2 Future Work

Future work would include the use of WordNet to allow non-lexical matching of script events, and developing a computational model that can allow and assess learning between cognitive scripts.

We would like to allow social agents to learn from cognitive scripts and they should be able to take the input from the voice of a person and construct the cognitive script. One way to achieve nontraditional/interesting new experiences from cognitive scripts can be through the use of conceptual blending.

References

Aamodt, A., & Plaza, E. (1994). *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. University of Trondheim.*

Albino, V., Garavelli, A., & Schiuma, G. (2000). A metric for measuring knowledge codification in organisation learning.

Bach, K., Althoff, K., Newo, R., & Stahl, A. (2011). *: A case-based reasoning approach for providing machine diagnosis from service reports. In: Ram, A., Wiratunga, N. Case-Based Reasoning Research and Development (Procs. of the 19th International Conference on case-based Reasoning).*

Bandura, A. (1977). *Social Learning Theory, Prentice Hall, Englewood Cliffs.*

Braziera, F., Dunin-Kepliczb, B., Treura, J., & Verbrugge., R. (1996). *Beliefs, Intentions and DESIRE, Vrije Universiteit Amsterdam. Department of Mathematics and Computer ScienceArtificial Intelligence Group De Boelelaan 1081a, 1081 HV Amsterdam,.*

Bredo, E. (1994). *'The social construction of learning', Handbook of academic learning: construction of knowledge, Academic Press, San Diego,.* (G. Phye, Ed.)

Carmel, D., & Markovitch, S. (1996). *Learnin Models for Intelligent Agents, Department of Computer Science.*

Chen., X. (2004). Scripts and Conceptual Change, In: P. Li, X. Chen, and H.X. Zhang (Eds.), In Science, Cognitive, and Consciousness,. 96-117.

CIPD. (2002). *How do people learn?* London.

Forbus, K., & Hinrichs. (2004). *Self-modeling in Companion Cognitive Systems: Current Plans.DARPA Workshop on Self-Aware Systems.* Washington, DC.

Gawish, M., Abbas, S., Mostafa, M. G., & A. B. M. Salem. (2013). *Learning cross-domain social knowledge from cognitive scripts, Proceedings of the 8th International Conference on Computer Engineering & Systems (ICCES), IEEE,.*

Gentner, D., & Smith, L. (2012). Analogical resoning, Encyclopedia of human behaviour, Elsevier. 130-136.

Glasersfield, E. V. (1989). *Learning as a constructive activity, Development in learning and assessment , Hodder and stoughton.* (P. Murphy, & B. Moon, Eds.) London.

Gros, C. (2010). In *Complex Adaptive Dynamical Systems, a Primer, Chapter 8 Elements of Cognitive Systems Theory.*

Hershberg, U., & Ninio, A. (2002). Cognitive systems and the special order of their environment.

Hodhod, R., Magerko, B., & Gawish, M. (2013). *Pharaoh: Context-Based Structural Retrieval of Cognitive Scripts. ADAM Lab, Georgia Institute of Technology, Georgia, USA. .* Cairo, Egypt.

Jacob, M., & Magerko, B. (2015). *Interaction-based Authoring for Scalable Co-creative Agents.*

Kimbrough, S. O., Wu, D., & Zhong, F. (2002). *Computers play the beer game: can artificial agents manage supply chains?*

Kirk, J., Mininger, A., & Laird, J. (2016). *Learning task goals interactively with visual.*

Kolodner, J. L., & Guzdial, M. (2000). *Theory and Practice of Case-Based Learning Aids.*

Langley, P., & Messina, E. (2005). Experimental Studies of Integrated Cognitive Systems.

Lazar, J., Heidi, F. J., & Hochheiser, H. (2010). *Research Methods in HUman-Computer Interaction.* Glasgow: Wiley.

Lintern, G. (2007). What is Cognitive System. *Proceedings of fourteenth international symposium aviation psychology*, 398-402.

Mahar, S., & Harford, M. (2004). *Research on Human Learning.* Department and education and training.

Mantaras, R. L. (2002). *Case-Based Reasoning. IIIA-Artificial Intelligence Research Institute CSIC-Spanish Scientific Research Council .*

Marchant, T. (2007). A meassurement-theoretic axiomatizaation of trapezoidal membership functions. *IEEE Transactions on Fuzzy Systems*, 5-6.

Mininger, A., & Laird, J. (2016). *Interactively Learning Strategies for Handling References to Unseen, Computing Science and Engineering, University of Michigan, Ann Arbor, MI 48103 USA.*

Moynihan, G. P., Jain, V., McLeod, R. W., & Fonseca, D. J. (2006). An expert system for financial ratio analysis. *International Journal of Financial Services Management*.

Negnevitsky, M. (2011). *Artificial Intelligence a Guide to Intelligent Systems - Thrid Edition.* Harlow: Pearson.

Negnevitsky, M. (2011). *Artificial Intelligence, A Guide To Intelligent Systems.* Harlow: Pearson.

Nelson., K. (1986). Event Knowledge: Structure and function in development scripts and narratives.

Newell, A., & H.A.Simon. (1972). *Human Problem Solving, Englewood Cliffs, NJ., Prentice Hall.*

Niedderer, H., & Schecker, H. (2010). Towards an explicit description of cognitive systems for research in physics learning.

Pelanek, R. (2015). Metrics for Evaluation of student Models, Masaryk University Brno.

Ratan, L., Iyer, R., & JAMS, E. (1988). *Similarity analysis of cognitive scripts.*

Salem, A.-B. M., & Gawish, M. (2016). *Study on Analogical Reasoning Methodologies For Developing Analogical Learning Systems. Department of Computer Science Faculty of Computer and Information Sciences Ain Shams University Cairo, EGYPT .*

Salem, A.-B. M., & Gawish, M. (2016). Study on analogy reasoning methododlogies for developing analogical learning systems, EGYPT.

Sauer, C. S., Rintala, L., & Roth-Berghofer., T. (2013). *Knowledge formalisation for hydrometallurgical gold ore processing. In: Research and Development in Intelligent Systems XXX, pp. 291–304.*

Sauer, C., Roth-Berghofer, T., Auricchio, N., & Proctor, S. (2013). *Recommending audio mixing workflows. In: Case-Based Reasoning Research and Development, pp. 299–313.*

Schalkoff, R. J. (2011). *Intelligent Systems, principles, paradigms, and pragmatics.* Jones and Bartlett Publishers.

Schank, R., & Abelson, R. (1977). Scripts, Plans Goals and Understandings: An inquery into human knowledge structures.

Shanks, D. R., & John, M. F. (2012). Characteristics of dissociable human leraning systems.

Appendix A: Cognitive scripts in SOAR

```
sp {apply*record-story*2
  (state <s> ^operator <op>
^next-story 2)
  (<op> ^name record-story)
-->
  (<s> ^next-story 3 2 - # remove ^next-story 2 and add ^next-story 3
      ^event |Audience buy ticket|
      ^next <e2>)
  (<e2> ^event |Audience enters theater|
      ^next <e3>
      ^next <e4>)
  (<e3> ^event |Audience buys popcorn/drinks|
       ^next <e4>)
  (<e4> ^event |Audience enter auditorium|
       ^next <e5>)
  (<e5> ^event |Audience sits down|
       ^next <e6>)
(<e6>^event |Audience watches movie|
      ^next <e7>
      ^next <e8>)
  (<e7> ^event |Audience is scared|
      ^next <e8>)
  (<e8> ^event |movie ends|
      ^next <e9>)
(<e9> ^event |Audience stand up|
       ^next <e10>)
  (<e10> ^event |Audience goes home|)

}
sp {apply*record-story*3
  (state <s> ^operator <op>
         ^next-story 3)
  (<op> ^name record-story)
-->
(<s> ^next-story 4 3 -
   ^event |Waiter asks how many people |
      ^next <e2>)
  (<e2> ^event |Customer answers one |
      ^next <e3>)
  (<e3> ^event |Waiter leads customer to table |
       ^next <e4>)
  (<e4> ^event |Customer sits down |
       ^next <e5>
   ^next <e6>)
  (<e5> ^event |Customer reads menu |
       ^next <e9>)
```

```
  (<e6> ^event |Waiter asks for drinks |
      ^next <e7>)
  (<e7> ^event |Customer orders drink |
      ^next <e8>)
  (<e8> ^event |Waiter delivers drink |
      ^next <e9>)
  (<e9> ^event |Customer order food |
      ^next <e10>
   ^next <e12>)
  (<e10> ^event |Waiter recommends wine |
      ^next <e11>)
  (<e11> ^event |Customer order wine |
      ^next <e12>)
   (<e12> ^event |Waiter takes order |
      ^next <e13>)
  (<e13> ^event |Chef cooks food |
      ^next <e14>)
  (<e14> ^event |waiter delivers food and wine |
      ^next <e15>
      ^next <e16>)
  (<e15> ^event |Customer eats food|
      ^next <e17>)
   (<e16> ^event |Customer drinks wine |
      ^next <e17>)
   (<e17> ^event |Customer asks for check |
      ^next <e18>)
   (<e18> ^event |Waiter gives customer check |
      ^next <e19>
      ^next <e20>)
  (<e19> ^event |Customer writes tip|
       ^next <e21>)
  (<e20> ^event |Customer signs name|
           ^next <e21>)
  (<e21> ^event |Customer leaves|)
}
sp {apply*record-story*4
  (state <s> ^operator <op>
         ^next-story 4)
  (<op> ^name record-story)
-->
(<s> ^next-story 5 4 -
     ^event |Audience put on formal attire|
     ^next <e2>^next <e3>)
  (<e2> ^event |Audience buy ticket at counter|
     ^next <e3>)
  (<e3> ^event |Audience show ticket|
     ^next <e4>)
  (<e4> ^event |Audience enter theater|
```

```
      ^next <e5>)
  (<e5> ^event |Audience sit on provided chair|
      ^next <e6>)
(<e6>^event |Audience listen to music|
     ^next <e7>)
  (<e7> ^event |orchestra stops performing|
     ^next <e8>
      ^next <e9>)
  (<e8> ^event |Audience applaud|
     ^next <e10>
     ^next <e11>)
 (<e9> ^event |Audience stand up|
      ^next <e11>)
  (<e10> ^event |Orchestra members go to backroom|)
   (<e11> ^event |Audience go home|)
}
sp {apply*record-story*5
  (state <s> ^operator <op>
       ^next-story 5)
  (<op> ^name record-story)
-->
(<s> ^next-story 6 5 -
   ^event |Customer orders medicine |
     ^next <e2>)
  (<e2> ^event |Pharmacist checks order |
     ^next <e3>
 ^next <e4>
  ^next <e8>)
  (<e3> ^event |Pharmacist recommends additional stuff |
     ^next <e5>
   ^next <e6>)
  (<e4> ^event |Pharmacist asks for prescription |
     ^next <e7>)
  (<e5> ^event |Customer orders additional stuff |
     ^next <e8>)
 (<e6> ^event |Customer declines offer |
     ^next <e8>)
        (<e7> ^event |Customer does not have prescription|
          ^next <e9>)
        (<e8> ^event |Pharmacist delivers medicine |
          ^next <e10>)
        (<e9> ^event |Pharmacist refuses to sell |
          ^next <e12>)
        (<e10> ^event |customer pays |
          ^next <e11>)
        (<e11> ^event |Pharmacist produces receipt |
          ^next <e12>)
        (<e12> ^event |Customer leaves|)
```

```
}
sp {apply*record-story*6
  (state <s> ^operator <op>
        ^next-story 6)
  (<op> ^name record-story)
-->
(<s> ^next-story 7 6 -
     ^event |Audience buys ticket|
     ^next <e2>)
  (<e2> ^event |Audience enters stadium|
     ^next <e3>)
  (<e3> ^event |Audience buys popcorn/drinks|
      ^next <e4>)
  (<e4>^event |Audience enter seating area|
      ^next <e5>)
  (<e5> ^event |Audience sits down|
      ^next <e6>)
   (<e6> ^event |Stand Up|
      ^next <e7>)
  (<e7> ^event |Audiunce goes home|)
}
sp {apply*record-story*7
  (state <s> ^operator <op>
         ^next-story 7)
  (<op> ^name record-story)
-->
(<s> ^next-story 8 7 -
      ^event |Audience buy ticket online|
      ^next <e2>)
  (<e2> ^event |Audience print ticket|
      ^next <e3>)
  (<e3> ^event |Audience show ticket|
      ^next <e4>)
  (<e4> ^event |Audience enter stadium|
      ^next <e5>
      ^next <e6>)
  (<e5> ^event |Audience buy snacks drinks|
      ^next <e6>)
  (<e6>^event |Audience enter seating area|
     ^next <e7>
     ^next <e8>
     ^next <e9>)
  (<e7> ^event |Audience sit on own chair|
     ^next <e10>
     ^next <e11>)
 (<e8> ^event |Audience sit on provided chair|
     ^next <e10>
     ^next <e11>)
```

```
(<e9> ^event |Audience stand|
      ^next <e10>
      ^next <e11>
      ^next <e12>)
 (<e10> ^event |Audience listen to song|
       ^next <e13>)
 (<e11> ^event |Audience sing along|
       ^next <e13>)
 (<e12> ^event |Audience dance|
       ^next <e13>)
(<e13> ^event |singers stop performing|
       ^next <e14>
      ^next <e15>)
(<e14> ^event |singers go to backroom|)
(<e15> ^event |audience go home|)
}
```

Appendix B: Cognitive scripts in myCBR

Script 1

audience_enters_theater; audience_buys_popcorn/drinks; audience_enters_auditorium; audience_sits_down; audience_watches_movies; Audience_is_scared; Movie_ends; Audience_stands_up; Audience_goes_home; na, na, na, na

Script 2

agent_talks_to_customer; agent_learns_about_customer; agent_recommends_a_car; customers_decides_to_buy_the_car; agent_tells_customer_price; customer_accepts_the_price; customer_pays_in_cash; customer_drives_car_home; customer_is_happy_about_car; na; na

Script 3

Audience_show_ticket; Audience_enter_theater; Audience_sit_on_provided_chair; Audience_listen_to_music; Orchestra_stops_performing; Audience_applaud; Audience_go_home; na; na

Script 4

Audience_show_ticket; Audience_enter_stadium; Audience_buy_snacks/drinks; Audience_enter_seating_area; Audience_sit_on_provided_chair; Audience_listen_to_song; Audience_dance; Singer_stops_performing; Audience_go_home; na; na

Script 5

Pharmacist_asks_for_prescription; Pharmacist_delivers_medicine; Customer_pays; Pharmacist_produces_receipt; Customer_leaves; na; na; na; na; na

Script 6

Waiter_asks_how_many_people; Customer_answers_one; Waiter_leads_customer_to_table; Customer_sits_down; Customer_reads_menu; Customer_orders_food; Waiter_takes_order; Chef_cooks_food; Waiter_delivers_food_&_wine; Customer_eats_food; Customer_asks_for_check; Waiter_gives_customer_check; Customer_writes_tip; Customer_leaves

Script 7

Audience_buys_popcorn; Audience_enters_seating_area; Audience_sits_down; Audience_watches_game; Audience_excited; Audience_go_home; na; na; na; na; na

Appendix C: Cognitive scripts in Pharaoh

<script>::="script(name("<script-name>"),parents(["
<script-parent>+"]),roots(["<script-root-node-id>+"]),actions(["<events>"]))";
<events>::=<event> | <event> "," <events>;
<event>::="node(id("<event-id>"),relation("<event-relation-name>"),attributes(["<attributes>"]),
children(["<child-node-id>*"]))";
<attributes>::=<attribute> | <attribute> ", " <attributes>;
<attribute>::="attribute("<attribute-name>", "<attribute-type>")"; and
<attribute-type>::="subject" | "object" |"adjective" | "preposition" | "conjunction".

script(name(cinema), parents([cinema]), roots([1, 11]), actions([node(id(1),
relation(buy),attributes([attribute(audience, subject), attribute(ticket, object)]), children([2])),
node(id(2),relation(enter), attributes([attribute(audience, subject), attribute(theater,
object)]),children([3, 4])), node(id(3), relation(buy), attributes([attribute(audience,
subject),attribute(popcorn/drinks, object)]), children([4])), node(id(4),
relation(enter),attributes([attribute(audience, subject), attribute(auditorium, object)]),
children([5])),node(id(5), relation(sit down), attributes([attribute(audience, subject)]),
children([6])),node(id(6), relation(watch), attributes([attribute(audience, subject),
attribute(movie,object)]), children([7, 8])), node(id(7), relation(is), attributes([attribute(audience,
subject),attribute(scared, adjective)]), children([8])), node(id(8),
relation(end),attributes([attribute(movie, subject)]), children([9])), node(id(9), relation(stand
up),attributes([attribute(audience, subject)]), children([10])), node(id(10), relation(go
to),attributes([attribute(audience, subject), attribute(home, object)]), children([0])),node(id(11),
relation(off), attributes([attribute(lights, subject)]), children([12])),node(id(12), relation(show),
attributes([attribute(trailers, subject)]), children([13])),node(id(13), relation(start),
attributes([attribute(movie, subject)]), children([6]))]))

I have submitted this thesis in partial fulfillment of the requirements for the degree of Master of Science

08/14/2017
Date

M. Praneetha
Praneetha Mandava

We approve of the thesis of Praneetha Mandava as presented here.

6/29/17
Date

Rania Hodhod
Rania Hodhod, Ph.D.
Assistant Professor, Thesis Advisor

8/14/17
Date

Shamim Khan, Ph.D.
Professor

8/17/2017
Date

Alfredo Perez, Ph.D.
Assistant Professor

8/14/17
Date

Aisha Patrice Adams, Ph.D.
Assistant Professor

8/14/2017
Date

Wayne Summers, Ph.D.
Distinguished Chairperson
Professor of Computer Science