Modeling affective system and episodic memory for social companions

Zhang, Juzheng

2017

Zhang, J. (2017). Modeling affective system and episodic memory for social companions.
Doctoral thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/72385>

<https://doi.org/10.32657/10356/72385>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

Modeling Affective System and Episodic Memory for Social Companions

BY

Zhang Juzheng

A THESIS SUBMITTED TO THE NANYANG TECHNOLOGICAL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING & INSTITUTE FOR MEDIA
INNOVATION
NANYANG TECHNOLOGICAL UNIVERSITY

2017

Abstract

Social companions, including virtual humans and social robots, have attracted increasing attention in recent years. The social companions interact with users in a social and human-like way that makes them believable and trustable. Recent studies have investigated the problem that people tend to feel bored and lose interest rapidly in the first few interactions with social companions. Thus, many studies have started to improve the capabilities of the social companions to preserve user engagement over long-term period. Such companions are required to display consistent personalities in interactions and remember what has happened in the past with particular users. Therefore, our research here focuses on designing the affective system and the episodic memory model, and studies how to integrate them compatibly with social companions to improve user engagement in interactions. Specifically, we have studied (1) how to design a general system architecture for social companions with necessary modules and connections; (2) how to design an affective system to express specific personalities by organizing emotional behaviors in particular patterns, (3) how to design an episodic memory model with more comprehensive similarity measure that can both improve the agent performance in tasks and the user engagement in social companion interactions, and (4) how to construct a benchmark for episodic memory models designed for free-dialog based social companion interactions.

Firstly, we propose a general system architecture for social companions to achieve engaging interactions. The architecture can be shared by our virtual human and our social robot. It is composed of three main components: perception, internal processing and action. Each component contains multiple interconnected modules. The affective system and the episodic memory are two important modules in this architecture which model human-like behaviors and intelligence. The communication and synchronization among the modules are deliberated in this work.

Secondly, we propose the personality-characterized mood dynamics (PCMD) model to achieve consistent emotional behavior patterns that reflect particular personalities of a

social companion. To connect static personalities and dynamic behaviors, we introduce the mood as an intermediate concept. With well-designed rules of mood-driven behaviors, we transform the problem of personifying behavior patterns to the problem of personifying mood dynamics. This is then formulated as an optimization problem, the objective function of which is constructed in a way that the overall mood converges to the personality after sufficient interactions.

Thirdly, we propose a mixed-correlation-analysis based episodic memory (MCAEM) model to improve memory retrieval accuracy. Specifically, we make analysis of correlations of memory elements by considering relations between elements, weights (importance) of attributes and order of events. An overall similarity measure is proposed based on the mixed correlation analysis. The proposed similarity measure can improve retrieval accuracy and thus further support higher-level capabilities. We validate the MCAEM model in two applications. First, the MCAEM model provides agents with better abilities of item finding, weapon selection and decision making which improve their performance in the Unreal Tournament 2004 (UT2004) game. Second, the proposed model has enhanced the user experience in social companion interactions in various aspects.

Finally, we construct a benchmark from movie scripts (BMS) to measure the retrieval performance of episodic memory models designed for free-dialog based social companion interactions. We build a dataset with 4,889 episodes that contain 38,058 attribute values and 76,123 events extracted from 106 movie scripts. Besides sentences in subtitles, our episodes also contain sequences of events constituted by attributes like subjects, objects, time, locations and emotion polarities. The benchmark is much more complicated compared to datasets in small domains as a result of considering words as additional attributes. To simulate real-world interactions, the retrieval cues are generated with noise and incompleteness. We analyze the consistency between the results of executing different models on our benchmark and on previous benchmarks. In addition, we also compare the proposed MCA model with two previous episodic memory models by evaluating them on our benchmark.

Acknowledgements

I would like to thank my supervisors Prof. Jianmin Zheng and Prof. Nadia Magnenat-Thalmann, for their time, effort, patience and constant support during this research. Their great knowledge and serious attitude to research work benefit me well.

I also want to express my gratitude for Prof. Edie Rasmussen, who is a professor from University of British Columbia and a visiting professor in NTU during the year 2014. She taught me a lot about information retrieval from scratch and discussed with me routinely every week. Without her valuable help and encouragement, I probably couldn't get through the tough time in my PhD.

Thank Dr. Budhitama Subagdja to provide me detailed explanation and source codes of the EM-ART model. In addition, I want to give my thank to my colleagues in IMI (Institute for Media Innovation), I have learned quite a lot from them in the four years. Especially, Dr Zerrin Yumak's work provided me a good starting point.

I appreciate the financial support for this research provided by SCSE (School of Computer Science and Engineering) and IMI, and the space and equipment provided by Institute for Media Innovation (IMI), Nanyang Technological University.

Deep thanks to my family and all my friends. In particular, I thank my parents for their encouragement and listening and my wife Panpan Cai for her support and love.

Contents

Abstract	i
Acknowledgements	iii
Table of Contents	v
List of Tables	ix
List of Figures	xi
1 Introduction	1
1.1 Background	1
1.2 Problem Statement and Objectives	3
1.3 Contributions and Thesis Organization	7
2 Literature Review	11
2.1 Social Companion	11
2.2 Affective System	16
2.2.1 Background in Psychology and Neuroscience	16
2.2.2 Components of Affective System	19
2.2.3 Computational Models	23
2.3 Episodic Memory	27
2.3.1 Background in Psychology and Neuroscience	28
2.3.2 Memory Requirements and Functionalities	30
2.3.3 Computational Episodic Memory Models	31

3	System Architecture for Social Companions	37
3.1	Our Social Companions	37
3.1.1	Our Social Robot	38
3.1.2	Our Virtual Human	39
3.1.3	Distinctions	39
3.2	Proposed System Architecture	40
3.3	Modules	42
3.3.1	Perception	43
3.3.2	Internal Processing	45
3.3.3	Action	47
4	PCMD: Personality-Characterized Mood Dynamics	51
4.1	Introduction	51
4.2	Related Work	53
4.3	Preliminaries	54
4.3.1	Psychology Concepts and the PAD space	54
4.3.2	Notation Declaration	55
4.3.3	Mood Dynamics of ALMA	57
4.4	Personality-Characterized Mood Dynamics (PCMD)	59
4.4.1	Rationale	59
4.4.2	Algorithms	60
4.4.3	Numerical Simulation	67
4.5	Experiments for Example Interactions	70
4.6	Case Study	74
4.6.1	Settings	75
4.6.2	Results and Discussion	76
4.7	User Study	77
4.7.1	Design and Procedure	78

4.7.2	Results	80
4.7.3	Discussion	82
5	MCAEM: Mixed-Correlation-Analysis based Episodic Memory	83
5.1	Introduction	83
5.2	Related Work	87
5.3	The Proposed Episodic Memory	92
5.3.1	Correlations for Relations Between Elements	93
5.3.2	Correlations for Weights of Attributes	98
5.3.3	Correlations for Order of Events	100
5.3.4	Overall Similarity Measure	102
5.3.5	Proposed Episodic Memory Model	103
5.4	Application on Intelligent Agents	105
5.4.1	Specific Adaptations for Intelligent Agents	105
5.4.2	Test on UT2004 Dataset	107
5.4.3	Test on UT2004 Platform	109
5.4.4	Discussion	114
5.5	Application on Social Companion	116
5.5.1	Specific Adaptations for Social Companions	117
5.5.2	Memory-Related Capabilities	119
5.5.3	User Study	121
6	BMS: A Benchmark from Movie Scripts for Episodic Memory Retrieval	129
6.1	Introduction	129
6.2	Acquisition of Dataset	131
6.3	Platform for Evaluation	132
6.3.1	Automatic Generation of Retrieval Cues	133
6.3.2	Evaluation Metric	135
6.4	Models for Comparison	136

6.4.1	ISAC Model	136
6.4.2	EM-ART Model	138
6.4.3	MCAEM Model	139
6.5	Experiments	139
6.5.1	Experiment Setup	139
6.5.2	Results	140
6.5.3	Discussion	144
7	Conclusions and Future Work	147
7.1	Contributions	147
7.2	Future Work	149
	References	153
	Author's Publications	168
	Appendix A: Questionnaire of User Study for PCMD Model	171
	Appendix B: Questionnaire of User Study for MCAEM Model	171

List of Tables

2.1	Integrated modules in previous social companion work	15
2.2	The dimensions of Big Five model.	19
2.3	Definitions for various memory subsystems	29
2.4	General functionalities of computational memory models [98]	31
3.1	Communication of modules of social companion	42
4.1	Mapping from 22 emotions to PAD [45]	55
4.2	Mapping from FFM to PAD [89].	55
4.3	Notations	56
4.4	Operators	56
4.5	Statistics of the ODC values	69
4.6	Three types of personality	71
4.7	The PAD settings of the three personalities	71
4.8	Experimental results of the user study	81
5.1	The parameters of EM-ART model in this implementation	92
5.2	All classes of attributes considered in the UT2004 domain	107
5.3	The means and standard errors of episode retrievals of the MCAEM model and EM-ART model	108
5.4	The learned locations for a few items	110
5.5	The retrieved weapon effects with given distances	110
5.6	Mehrabian mood types.	118

5.7	Questions for retrieval performance	124
5.8	Results for objective measure and the subjective user study	126
5.9	Pearson's r for correlations of aspects of user experience and the subject impression for memory	126
6.1	The parameters of EM-ART model on the proposed BMS benchmark . . .	139
6.2	Comparing the scales of the tested datasets	140
6.3	Evaluating the event retrieval of models on the three benchmarks. The nor- malized MRR (nMRR) measure is adopted to compare the relative perfor- mance of models on the three benchmarks	141
6.4	Evaluating the episode retrieval of models using different datasets. The normalized MRR (nMRR) measure is adopted to compare the relative per- formance of models on the three benchmarks	141

List of Figures

1.1	Technical components of my PhD research	4
2.1	Examples of anthropomorphic companions	12
2.2	Examples of social companions: (a) Robot geographical teacher Eva, (b) Social companion for health and fitness, (c) Virtual job interviewer (d) Virtual companion to recommend recipes (e) Virtual companion to discuss photographs, (f) Virtual human to discuss movies	13
2.3	The components of an affective system	20
2.4	The cognitive structure of OCC structure [118]	22
2.5	Case study from Orozco et al. [99]. The left and right figures show the reactions of a hysteric and a depressive virtual human for a very negative event respectively	25
2.6	Case study from Becker et al. [8]. The virtual human gets angry and leaves the screen if the user keeps insulting him	26
2.7	Relations of memory subsystems	29
2.8	Anatomical human brain components related to episodic memory	30
2.9	Example testing environments for episodic memory models designed for intelligent agents. (a) is the TankSoar game used in Nuxoll and Laird [98], (b) is the dynamic environment used in Ho et.al [61]	33

2.10	Example testing environments for episodic memory models designed for social companions. (a) is the FearNot story-telling game used in Dodd [33], (b) is the robot tutor Eva used in Kasap et.al. [70]	34
3.1	The Nadine social robot	38
3.2	The Sophie virtual human	39
3.3	The system architecture of our social companion	40
3.4	The diagram of face recognition system [107]	44
3.5	The diagram of upper body gesture recognition [139]	45
3.6	The implemented behavior-tree planner	46
3.7	The diagram of dialog manager	48
3.8	Example gestures of Nadine	49
3.9	Example gestures of Sophie	50
4.1	The rationale of the PCMD model for social companion	52
4.2	The schema of affect dynamics	61
4.3	The results of the cosine values between the current mood and the personality in the first test	69
4.4	Mood dynamics for an anger emotion	72
4.5	Mood dynamics for a gratitude emotion	72
4.6	Mood dynamics for an anger emotion followed by a gratitude emotion . . .	73
4.7	Mood dynamics for a gratitude emotion followed by an anger emotion . . .	74
4.8	The system structure of the virtual human	75
4.9	Sophie interacts with polite user and rude user.	77
4.10	Interaction with Nadine with the PCMD model or the ALMA model	78
4.11	Comparison performance in emotion sensitivity	79
4.12	Comparison performance in recognition easiness	80
4.13	Comparison performance in personality polarity	81

5.1	The four aspects of the similarity measure in episodic memory	84
5.2	Examples for two types of attribute-attribute relations	85
5.3	Examples of element weights in particular contexts	86
5.4	The overall structure of EM-ART model	90
5.5	The structure of a single ART channel	91
5.6	The pipeline to build the hierarchical relational space	93
5.7	The first two dimensions of the event space in the UT2004 domain. The blue points and green circles represent parts of attributes and event clusters respectively	94
5.8	The first two dimensions of the episode space in the UT2004 domain. The green circles and red crosses represent parts of event clusters and episode clusters respectively	95
5.9	The retrieval by similarity with relations	98
5.10	The sliding window method to analyze event-order correlations of episodes	100
5.11	The combination of algorithms for aspects of similarity measure	102
5.12	The framework of episodic memory	104
5.13	The mechanism of short-term and long-term memory	105
5.14	The membership functions for enemy distance	106
5.15	Comparing episode retrieval of the MCAEM model and EM-ART model on the UT2004 dataset for twenty trials. The blue and red curves represent the MCAEM model and EM-ART model respectively	108
5.16	The kill/killed rates of tested agents against the <i>EMART-1800</i> agent	115
5.17	Event structure with attributes for social companion-user interactions	117
5.18	Some participants	122
5.19	Effects of the memory component	125
6.1	An example part of movie script. The attributes to be extracted are denoted in read boxes and words.	131

6.2	An example episode extracted from movie scripts	133
6.3	The workflow for the evaluation platform	134
6.4	The computation of fingerprint in ISAC model	137
6.5	Results of event retrieval given multiple levels of erroneoususness. The ratio of partial cues is set to 70%, 50% and 30%, and the threshold of noisy cues is set to 0.9, 0.7 and 0.5. The blue and red colors stand for the MCAEM and EM-ART model respectively.	142
6.6	Results of event retrieval given various cues when the ratio of partial cues is set to 70% and the threshold of noisy cues is set to 0.7. The blue and red colors stand for the MCAEM and EM-ART model respectively.	143
6.7	Results of episode retrieval given multiple levels of erroneoususness. The ratio of partial cues is set to 70%, 50% and 30%, and the threshold of noisy cues is set to 0.9, 0.7 and 0.5. The blue, red and green colors stand for the MCAEM, EM-ART and ISAC model respectively.	144
6.8	Results of episode retrieval given various cues when the ratio of partial cues is set to 70% and the threshold of noisy cues is set to 0.7. The blue, red and green colors stand for the MCAEM, EM-ART and ISAC model respectively.	145
7.1	Page 1 of questionnaire	171
7.2	Page 2 of questionnaire	172
7.3	Page 3 of questionnaire	173
7.4	Page 4 of questionnaire	174
7.5	Page 1 of questionnaire	175
7.6	Page 2 of questionnaire	176
7.7	Page 3 of questionnaire	177
7.8	Page 4 of questionnaire	178
7.9	Page 5 of questionnaire	179

Chapter 1

Introduction

1.1 Background

Social companions refer to “personalized, multi-modal, helpful, collaborative, conversational, learning, social, emotional, cognitive and persistent computer agents that interact with the user over a long period and build relationship with the user” [24]. There are considerable amount of research on social companions in recent years. For example, the Siri from Apple and the Cortana from Microsoft work as personal assistants on computers or smart phones, which are special cases of social companions. However, this thesis mainly focuses on the companions with realistic human figures including virtual humans and social robots. Generally speaking, social companions should have the following characteristics [24]: (1) conversational abilities, including cognitive abilities, emotional competence, social-cultural competence and natural language understanding; (2) adaptability to user preferences; (3) utility (usefulness); and (4) long-term interaction.

In particular, maintaining long-term engagement one important and challenging issue in social companion studies. User interests to the state-of-art social companions decrease rapidly after a few interactions [76]. Therefore, many research projects are aimed at pre-

serving user interest and engagement over the long-term (e.g., BeingThere^{*}, EVA[†], LIREC[‡], Companions[§], SERA[¶], CompanionAble^{||} and ALIZ-E^{**}). A promising solution to slowing down the decrease of user interests is to have social companions maintain consistent personalities, remember and learn from past experiences, develop individual styles of communications and adapt to user preferences. Thus researches on episodic memory and personalized affective system become necessary [70, 76].

Affective systems enable social companions to think, express and communicate emotionally with users. This is achieved by updating companions' affect states to emotion appraisals and supporting emotional decision making using these states. Many previous studies focus on affect detection, affect appraisal, affect dynamics and affect response [16, 52, 37]. Both affect detection and appraisal deal with the input of affects. The affect dynamics updates the affect states from time to time, and the affect response accounts for the emotional decision making. In particular, the affect detection processes physical, machine-detectable signals from video [32, 21] or audio [75, 29] sensors to abstract the semantic meanings. Then the detected environmental inputs are appraised to generate appropriate emotions with intensities [101, 43, 73, 112]. The resulting emotions update the affect states of the companions according to the contexts of interactions [45, 35, 70]. Finally, the emotional decision making enables social companions to make appropriate responses to users [37, 108, 94]. Unfortunately, most previous studies have problem in maintaining consistent, personalized affect features over long period. In fact, how to achieve long-term personalities is still an open problem. The term "long-term personality" refers to that the personality should characterize dynamics of affect states and shape consistent individual behavior patterns over the long term.

The purpose of episodic memory is to encode, store and retrieve [98] past experience of

^{*}<http://imi.ntu.edu.sg/>

[†]<http://www.miralab.ch/>

[‡]<http://lirec.eu/>

[§]<http://www.companions-project.org/>

[¶]<http://project-sera.eu/>

^{||}<http://companionable.net/>

^{**}<http://www.aliz-e.org/>

social companions or intelligent agents. The stored episodes are composed of autobiographical events described by attribute values and temporal-spatial relations to other events [127]. Computational episodic memory models play important roles in social companions and intelligent agents. On one hand, episodic memory can improve user engagement in social companion interactions. It has been applied in various domains such as education [53, 70], health and fitness [128], job interview [46], museum [102] as well as communication about user interested topics like photos [137], movies [80], traveling [18] and musics [116]. On the other hand, episodic memory can support multiple cognitive capabilities such as virtual sensing, reasoning, learning and decision making [98]. Many studies have demonstrated the usefulness of episodic memory in improving task performance [98], enhancing agent survivability [61] and reporting key information [14].

Although prior studies have achieved interesting and exciting results, many challenges are still there for the purpose of engaging interactions of social companions.

1.2 Problem Statement and Objectives

In this research, we focus on modeling affective system and designing episodic memory for social companions that have realistic human figures, such as virtual humans and social robots. As shown in Figure 1.1, there are several technical components: the system architecture for social companions that can integrate episodic memory and affective system; the affective system model that can generate long-term personalized behavior patterns; the episodic memory model which encode, store, retrieve and learn past experience; and the benchmark for episodic memory models for free-dialog based companion interactions.

System Architecture for Social Companions Social companions have been getting increasing focus as a multidisciplinary research. They can interact with people as capable assistants, knowledgeable tutors, considerate doctors, smart companions and loyal friends. A social companion has to integrate many modules to achieve autonomous and human-like

Figure 1.1: Technical components of my PhD research

interactions. To design social companions, we need to consider the following questions: (1) what are the main functionalities they should have, (2) what modules should be included to support these functionalities, (3) how these modules fit into the system and connect with the others, (4) how an efficient message exchanging mechanism should be designed, (5) how to combine these functionalities to achieve high-level capabilities, (6) how to enable easy integration and removal of modules, and (7) especially how to integrate the affective system and episodic memory in the system architecture.

Affective System Modeling Given a predefined repository of emotional behaviors, we need to design an affective system that automatically combines these emotional behaviors in consistent behavior patterns to express particular personalities.

Although consistent behavior patterns are necessary to convey the personalities of social

companions, it doesn't mean that behavior patterns should remain unchanged in whatever situations [15]. For believable interactions, we need to adapt the reactive behaviors to different situations and keep the overall effects of behavior patterns consistent after sufficient interactions. For example, although an extrovert social companion is usually talkative and eager to communicate with users, he could keep salient on solemn occasions. His personality would be identified as extrovert from his overall behavior patterns regardless of particular occasional behavior.

The challenge is that instead of manually designing behavior patterns for specific personalities directly, an intermediate concept is needed to automatically make the behaviors of the social companions consistent with personalities, as well as flexible enough to generate appropriate responses for various situations. Besides the personality (the unchanged nature of affects), emotions and moods are parts of the affective models of social companions too [45]. The emotions and moods have big differences in effects. Emotions, which are short-term affects, influence instant behaviors including facial expressions and gestures, while moods, which are long-term affects, influence longer-term behavior patterns including dialog strategies and decision-making [12]. It is thus reasonable for us to treat moods, instead of emotions, as the bridge between personalities and behavior patterns so that behavior patterns are influenced by mood dynamics characterized by personalities. By doing so, the problem of personalizing behavior patterns is transformed to the problem of personalizing mood dynamics. However, this effect has been rarely explored by previous studies on computational affective systems.

Episodic Memory Modeling Episodic memory modeling requires to efficiently encode, store, retrieve and learn from past experiences. A general episodic memory model should capture the snapshots of all elements from the working memory and make preparations for future tasks.

Memory retrieval serves as fundamental supports for high-level capabilities. Episodes or events are retrieved according to their similarities with retrieval cues. To improve the

accuracy of the retrieval, we need to make correlation analysis of memory elements (attributes, events and episodes). The correlation analysis considers the relations between elements, the weights of attributes and the order of events. These three types of correlations are considered by a few prior works, but they still need to be improved in following directions.

Firstly, it is very important to consider the relations between memory elements in the same levels or across different levels. Prior studies have considered attribute-event and event-episode relations, however the relations in the same levels (attribute-attribute, event-event and episode-episode) are rarely modeled or utilized for memory organization or retrieval. Secondly, instead of treating each attribute with equal importance, attributes should be given different weights at particular contexts. Some prior studies define the weights of attributes based on their frequency of occurrence. However the specialness of attributes at particular episodes has rarely been considered. Thirdly, the order of events are often ignored in earlier work. It can be very useful to differentiate episodes with similar contents. Some recent studies include the order of event in their similarity measures, but these algorithms can be better optimized for episodic memory.

A Benchmark for Episodic Memory for Free-Dialog based Companion Interactions

To the best of our knowledge, there is no benchmark designed for the episodic memory models aimed at free-dialog based social companion interactions. Many previous studies evaluate their episodic memory models by conducting user studies. The results of user studies are subjective and hard to be reproduced and compare. It is also difficult to get user feedbacks on the correctness of episode retrievals. Most episodic memory models for intelligent agents are evaluated by measuring their retrieval performance on datasets such as Logistics [122], Monroe plan [66] and UT2004 [135]. Nevertheless, these datasets come from smaller domains that are not complex enough to evaluate episodic memory models in the context of free-dialog based companion interactions. Our research is aimed at building a benchmark evaluation for this complex domain. However, it is difficult to collect large

among of episodes during interactions. Moreover, we also need to automatically generate proper retrieval cues to simulate noisy, ambiguous and incomplete inputs from users in free dialogs.

1.3 Contributions and Thesis Organization

This thesis has made several contributions to the research of affective system and episodic memory for social companion interactions. In particular, we have developed four technical components: a general system architecture integrating modules including affective system and episodic memory, a personality-characterized mood dynamics (PCMD) model in the affective system, a mixed-correlation-analysis based episodic memory (MCAEM), and a benchmark designed for episodic memory aimed at free-dialog companion interactions. More details are given below:

- We propose a general system architecture for social companions which is shared by our virtual human Sophie and social robot Nadine. It is composed of three main components: perception, internal processing and action. Each component contains multiple interconnected modules. We design an efficient mechanism that optimize the communication, synchronization, and easy integration and removal of the modules. Particularly, we embed the affective system and episodic memory into the system architecture to preserve user interest and engagement in companion interactions over the long term.
- To enable social companions to express various personalities given fixed repository of available behaviors, we transform the problem of personalizing behavior patterns into personalizing mood dynamics. We propose a personality-characterized mood dynamics (PCMD) model, which guarantees that the tendency of the mood is consistent with the personality after sufficient interactions. The convergence effect is achieved by correctly weighting different types of emotions in the emotion-mood interactions.

Finding the emotion weights is formulated as a convex optimization problem. Our numerical simulations, case studies and user studies demonstrate the effectiveness of the PCMD in endowing companions with consistent behavior patterns for specific personalities.

- To improve memory retrieval performance, we propose a mixed-correlation-analysis based episodic memory (MCAEM). Our work makes novel analysis of correlations of memory elements (attributes, events and episodes) by considering the relations between elements, the weights (importance) of attributes, and the order of events. For the relations between elements, we construct a hierarchical relational space that maps every memory element as a vector and computes their relations using distances. In particular, two kinds of attribute relations are considered in this work, the *attribute correlations between classes* and the *attribute similarities within classes*, which especially benefit retrievals with partial and noisy cues. To analyze the weights of attributes, we apply the TF-IDF (term frequency-inverse document frequency) method [110] to include both the occurrence frequency and the specialness of attributes in particular episodes. For the order of events, we propose a sliding-window analysis which is balanced and robust to positions of events. Finally, based on the mixed correlation analysis, we propose an overall similarity measure which improves memory retrieval accuracy. The proposed MCAEM model is validated to be effective in both enhancing various aspects of user experience in companion interactions and improving agent performance in particular tasks.
- We build a benchmark from movie scripts (BMS) to evaluate the retrieval performance of episodic memory models in the context of dialog-based companion interactions. The dataset is built by extracting 4,889 episodes, 76,123 events and 38,058 attribute values from 106 movie scripts. We adopt the mean reciprocal rank (MRR) to measure the retrieval performance. Four types of retrieval cues (complete cues, partial cues, noisy cues and noisy partial cues) are generated to simulate users' noisy and

incomplete inputs during interactions. We provide experiments to test two types of performance of various episodic memory models in event and episode retrievals: the average retrieval accuracy and the average retrieval time. We analyze the consistency between the results of executing different models on our benchmark and on previous benchmarks. In addition, we also compare the proposed MCAEM model with two previous episodic memory models by evaluating them on our benchmark.

With the proposed affective system and episodic memory, a social companion can make appropriate emotional responses, remember and learn from past experience, and behave in a human-like and believable way with consistent personalities.

The remainder of the thesis is organized as follows:

- Chapter 2 reviews related work on social companions, affective system and episodic memory.
- Chapter 3 presents the proposed system architecture for social companions as used in our virtual human Sophie and social robot Nadine. The modules in the framework are introduced.
- Chapter 4 presents the personality-characterized mood dynamics (PCMD) model with numerical simulations, case study and user study designed and conducted for evaluation.
- Chapter 5 describes the mixed-correlation-analysis based episodic memory (MCAEM). The MCAEM model is evaluated by performance improvements on intelligent agents and enhancements of user engagement in social companion interactions.
- Chapter 6 presents a benchmark for the episodic memory models designed for free-dialog based companion interactions. Using this benchmark, we have also compared our episodic memory with two other models in various aspects.
- Chapter 7 summarizes the thesis and discusses some directions for future research.

Chapter 2

Literature Review

In this chapter, we review previous studies on social companions, affective system and episodic memory. The review includes both psychology and neuroscience theories and computational models.

2.1 Social Companion

Social companions have received much attention in recent years. Danilava [24] defined social companions as “personalized, multimodal, helpful, collaborative, conversational, learning, social, emotional, cognitive and persistent computer agent that knows its owner, interacts with the user over a long period of time and builds a (long-term) relationship to the user.”

Many anthropomorphic companions have been developed, such as the Sony AIBO dog [41], PARO sea lion [133] and iCat [77] as shown in Figure 2.1. A detailed review can be found in Leite et.al [76]. Different from the work on anthropomorphic companions, our thesis focuses on humanoid social companions, such as virtual humans and social robots. It is more challenging to build humanoid companions than anthropomorphic ones because users have higher expectations due to their human-like appearance. Thus complicated human-like behavior, cognition and communication mechanisms are expected for

humanoid companions.

Figure 2.1: Examples of anthropomorphic companions

Many research projects are aimed at building realistic humanoid companions and exploring social companion-user interactions, such as BeingThere^{*}, Eva[†], LIREC[‡], Companions[§], SERA[¶], CompanionAble^{||} and ALIZ-E^{**}. Some studies have reported that users tend to rapidly lose their interest for social companions after the first few interactions [49, 109, 70]. The rapid decrease of interest results from repeating scripted behavior patterns and limited awareness of past experience and current context. How to maintain user engagements over long period is still an open issue. Therefore, this thesis is aimed at endowing social companions with long-term predictable personalities as well as a better memory associating past experiences to improve and prolong user engagements.

As shown in Figure 2.2, social companions have many applications such as education, health caring, job interview, public service as well as communication for user interested topics such as photos, movies, travel and musics. In this section, some social companion work in the state-of-art are presented below, organized by their application categories.

For the application on education, Kasap et.al [70, 68] develop a virtual human and a social robot, Eva, to teach users geographic knowledge. Eva identifies users and recognize their speech by face recognition and speech recognition modules respectively. She

^{*}<http://imi.ntu.edu.sg/>

[†]<http://www.miralab.ch/>

[‡]<http://lirec.eu/>

[§]<http://www.companions-project.org/>

[¶]<http://project-sera.eu/>

^{||}<http://companionable.net/>

^{**}<http://www.aliz-e.org/>

Figure 2.2: Examples of social companions: (a) Robot geographical teacher Eva, (b) Social companion for health and fitness, (c) Virtual job interviewer (d) Virtual companion to recommend recipes (e) Virtual companion to discuss photographs, (f) Virtual human to discuss movies

makes decisions with a hierarchical task network (HTN) planner and organize dialogs by a finite-state-machine (FSM)-based dialog manager. By integrating an emotion module and a database-like memory module, Eva can generate dialogs based on predefined scripts with adaptations to her current emotional states and the recalled user previous performance. Meanwhile, appropriate facial expressions and gestures are displayed too. However, this work is limited to a small specific domain in which all tasks, rewards and dialog patterns are predefined.

For the application on health caring, Turunen et.al [128] build three conversational companions to work collaboratively. They are a *Physical Home Companion* to keep users healthy lifestyle and remind users activity plans, a *Mobile Companion* to record user outdoor activities by tracking their exercise progress and finally giving a report, and a *Virtual Cooking Companion* to recommend daily recipes. Similar to Kasap et.al [70, 68], this work integrates a dialog manager with domain specific patterns, a cognitive manager with HTN planner and natural language processing tools for understanding.

For the application on job interview, Gebhard et.al [46] design two virtual characters to train people to get through stressful interviews. The two virtual characters have opposite behavior patterns: understanding and demanding. The understanding interviewer interacts with users warmly and friendly to enhance their self-efficacy, while the demanding interviewer behaves toughly and unfriendly to train users for difficult situations with pressure. The behavior patterns of the virtual interviewers are expressed by manually designed verbal and non-verbal behaviors for the two specific personalities.

For the application on public services, Papangelis et.al [102] design a virtual agent as guider to introduce to the public the exhibitions in a museum. The agent integrates an adaptive dialog system to learn optimal dialog policies using online reinforcement learning strategies. The agent searches information in a database manner and learns from rewards for each action. Different with traditional dialog systems that are “static and inflexible to individual user needs or to changes in their environment”, The proposed adaptive dialog system can rapidly change to learned user preferences. However, this agent lacks other necessary capabilities as a social companion except for the dialog-based communication. Another example is the work of Cassell et.al [20], which designs an embodied conversational virtual agent, Rea, to introduce real states to users. Displayed in a project screen, Rea can analyze user inputs from speech and vision and make dialog responses with gestures and facial expressions. She can handle mixed initiatives, interruptions and turn-takings in conversations as well.

For the application on communication for user interested topics, Adam et.al [1] develop a conversational toy companion to build engaging interactions and long-term relations with children. To achieve non-task-oriented conversation, the toy companion integrates simple memory functionalities. The companion collects personal information of these children by asking them explicitly at right time and uses these information in dialog. In addition, many available sentences used in the companion-user conversations are derived from corpora of children dialogs. Another sympathetic companion is designed by Pulman et.al [104] to discuss with users about their daily works and to put positive influence on their attitudes using

Table 2.1: Integrated modules in previous social companion work

Companions	Face Recognition	Speech Recognition	Planner	NLP	Affective System	Episodic Memory	Dialog Generation	Gesture Generation
Kasap et.al [70, 68]	✓	✓	✓	limited	✓	✓	✓	✓
Turunen et.al [128]	✓	✓	✓	✓	No	No	✓	✓
Gebhard et.al [46]	✓	✓	✓	limited	✓	No	✓	✓
Papangelis et.al [102]	No	✓	✓	limited	No	No	✓	No
Cassell et.al [20]	✓	✓	✓	limited	No	No	✓	✓
Adam et.al [1]	No	✓	✓	✓	No	✓	✓	No
Pulman et.al [104]	No	✓	✓	✓	✓	✓	✓	✓
Liu et.al [80]	✓	✓	✓	✓	✓	No	✓	✓

affective dialog strategies. To support free conversations rather than task-oriented ones, the proposed memory collects key events by extracting them from user speech using the information extraction (IE) and natural language processing (NLP) techniques. The dialog strategies and facial expressions are selected accordingly to the emotions recognized from user speech. The companion will make responses with recalled memories to the detected obvious differences in user affects between their current narratives and the previous ones for the same events. In addition, a virtual human, Sophie, designed by Liu et.al [80] can discuss with users about their favorite movies. Useful knowledge about users and movies are extracted from their Facebook homepages. Sophie integrates many modules including face recognition, speech recognition, a behavior-tree based planner and dialog manager with predefined patterns. She generates dialogs by filling the patterns with the relevant information about users and movies as well as their recent moods detected from the texts on their Facebook homepages.

The above social companions are built with complicated systems. These systems usually share perception modules (such as face recognition and speech recognition, etc.), processing modules (such as planner, affective system and episodic memory, etc.) and reaction modules (such as generator for dialog manager, facial expressions and gestures, etc.). Table 2.1 displays the integrated modules in the above social companions. Some companions have integrated the affective system (including emotions, moods and personalities) and episodic memory for past experience. The integration of these two modules have enhanced the companions towards engaging interactions [70, 68]. In the following two sections, we

review theories and computational models for affective system and episodic memory for social companion-user interactions.

2.2 Affective System

The study of human affects in psychology and neuroscience has a long history. These theories of human affects have inspired many computational affective models. In this section, we first introduce definitions and principles of human emotions, moods and personalities in psychology and neuroscience. Then we review the inspired computational models and evaluation methods for affective systems.

2.2.1 Background in Psychology and Neuroscience

Human affects can be classified into emotions, moods and personalities. The emotions and moods are described as affective states changing from time to time, and the personalities are described as affective traits which represent the innate characteristics of human affects.

Emotion

Emotions are immediate subjective responses to events in human interactions with environment. The fundamental characteristics of emotions are follows. Firstly, emotions are object-directed. They are usually related to particular objects or situations [44]. Secondly, emotions are relatively short-lived. Thirdly, emotions prepare people for immediate reactions, such as facial expressions and gestures [26].

Based on the work of LeDoux et.al [74] in neuroscience, the thalamus, limbic system, and cortex are three important brain regions related to human emotions. The thalamus collects and processes all perception information from external environment. The processed information is then sent to the cortex and limbic system. The limbic system generates corresponding emotions by analyzing relations of the processed information with the current

needs and goals. The generated emotions prepare human body for physiological responses as well as regulate cognitive processes, such as attentions, in the cortex.

One of the most fundamental effects of emotions is to capture attention. Triggered emotions drive people to pay attention to the related events. The degree of such attention as well as the emotion arousal are determined by event importance in the context of current goals [22]. The attention is further influenced by a secondary process of emotion regulation. Once a triggered emotion is appraised as undesirable, higher cognitive processes will shift human attention away from the related event [54]. Besides the effects on attention, emotions also have great influence on human memory. Thorson and Friestad [125] propose that events with higher emotion arousal are better recalled from episodic memory. Moreover, undesirable events are better remembered than desirable events [97]. In addition, emotions urges memory to focus on main contents of events while neglecting unimportant background details [60].

Mood

Moods are general, diffuse and global affective states staying over long period of time. Different from emotions, moods have no direct relation to specific objects or events. For example, “A person can be sad about something (an emotion) or generally depressed (a mood)” [12]. Meanwhile, moods are long-lived affective states that influence human cognitive strategies and processing [26]. In addition, moods and emotions have mutual influence on each other. Moods can strengthen those emotions which have similar properties (valence, arousal, dominance) to themselves. Emotions, on the other hand, drive mood dynamics [12].

Moods have great influence on human attention, cognition, behaviors and memories. Some of these relations are considered in our work. Firstly, human attentions are influenced by mood-congruent information. According to Bower [11], “a person selectively attends to and processes information that is congruent with his current mood”. For example, people tend to be more sensitive to negative aspects of events if they happen to have bad moods,

and vice versa. Our mood dynamics model in Chapter 4 has implemented this tendency in a manner that the intensities of new emotion inputs are adapted by their correlations with the current mood on the appraisal stage. For example, a “depressed” mood can strengthen a “shame” emotion while weaken a “happy” emotion.

Secondly, human cognitive strategies and behavior patterns are influenced by moods. Moods can change human style of information processing. As claimed by Andrade [6], positive moods bias broader/top-down processing, while negative moods bias narrower/bottom-up processing. According to Isen and Means [63], moods adapt human cognitive strategies such as helping others or taking risks in a way that human in positive moods tends to be altruistic in appropriate degree. In addition, positive moods can also decrease unreasonable decisions with high risk to preserve its positiveness. This tendency is considered by our mood dynamics model in Chapter 4 that our companions tend to take differing behavior responses based on the current moods even for the same emotion inputs.

Thirdly, moods also take influence on human memory with the mood-congruency effects and mood-dependent effects. The mood-congruence effect is that events are better remembered if the current moods match the emotional contents in the events [38]. The probabilities of correct recalls correlate with the strength of the mood-congruence effect at the event encoding [39]. Environmental information consistent with the current mood is more likely to be noticed, to be associated with other facts about that mood, and to promote elaborations to embellish its meaning [55]. On the other hand, the mood-dependence effect [38] is that events are more reliably recalled if the current mood is similar to the mood when they were encoded. An analogy is that our feelings are like a magnet that selects the iron filings from a heap of dust, drawing to itself whatever incoming material it can use [11]. Thus our memory model in Chapter 5 considers moods of social companions as attributes to be encoded and recalled.

Personality (Temperament)

Personalities are predispositions of human affects that determine tendency of moods [12]. A popular personality model is the “Big Five” or called “Five Factor Model” (FFM) [50]. This model figures out five fundamental factors of personalities based on a large amount of self-reports, peer-reports and factor analytic studies. As shown in Table 2.2, the five factors are: extroversion, agreeableness, neuroticism, openness, and conscientiousness.

Another related concept is temperament which refers to “the characteristic phenomena of an individual’s emotional nature” [4]. Temperament and personality are related and similar concepts. Significant similarities have been found between temperamental dispositions and the Big Five trait factors [3, 89]. Specifically speaking, personality is believed to be developed on the basis of temperament. In affective systems for social companions, temperament and personality can be almost mixedly used.

Table 2.2: The dimensions of Big Five model.

Personality Dimension	Descriptive Words
Openness (O)	imaginative, intelligent, and creative
Conscientiousness (C)	responsible, reliable, and tidy
Extroversion (E)	outgoing, sociable, and assertive
Agreeableness (A)	trustworthy, kind, and cooperative
Neuroticism (N)	anxious, nervous, and depressive

2.2.2 Components of Affective System

Human affects (including emotions, moods, personalities) play important roles in social interactions. Social companions are expected to understand user affects in interactions, update their own affective states accordingly, and finally generate natural behavior responses as results. These affective capabilities make social companions more human-like, trustful and appealing to users [19].

An affective system endows social companions with an affective intelligence to make reasonably and affectively appropriate responses to event stimuli, and to communicate with

users in an autonomous and believable way. As shown in Figure 2.3, an affective system usually contains five components: **affect detection**, **affect appraisal**, **affect dynamics** and **affect response** [16, 52, 37].

Figure 2.3: The components of an affective system

The **affect detection** obtains emotions from video or audio sensors. The instruments, such as cameras or Kinects, build the visual perception of social companions, based on which technologies such as affect recognitions from facial expressions [36, 87], postures [23, 32] or gestures [56, 21] of users are developed. Meanwhile, microphones or other acoustic sensors are used to recognize emotions from user speech based on acoustic characteristics [75, 29] or texts of speech [114, 5, 25, 120]. The texts are usually preprocessed by speech recognition and natural language processing. The recognized emotions from multiple sensors should be fused to keep consistent [113, 21].

The **affect appraisal** cognitively transfers abstract events and appraisal variables into emotions with intensities. Different from the concepts of emotions used in our daily life, the

types of emotions just represent combinations of appraisal variables. According to Marsella et.al [85], there are four main appraisal models [43, 112, 73, 101] considered in previous work. Frijda [43] differentiates emotion classes according to 14 appraisal variables and 18 kinds of action readiness. Scherer [112] identifies emotions based on six appraisal variables: suddenness, goal relevance, intrinsic pleasantness, conduciveness, control and power. Lazarus [73] separates appraisal process into the primary appraisal and secondary appraisal. They adopt six appraisal variables: motivational relevance and congruence, accountability, problem-focused coping potential, emotion-focused coping potential and future expectancy. EMA [52] is an example of affective system that is inspired by the Lazarus's model. The EMA model provides a general computational framework for emotional appraisal and coping for intelligent agents. Although the above three appraisal models have been adopted by some prior work [117, 117, 9], the OCC model [101] is the most popular one that is widely used in affective computing [52, 37, 31, 45]. As shown in Figure 2.4, the OCC model specifies 22 types of emotions according to agent positive or negative reactions to three aspects of events: consequence, actions, and objects. More specifically, it considers (1) the desirability of consequence of events according to current goals (desired states of the world), (2) the praiseworthiness of actions of agents according to standards (ideas about how people should act) and (3) the agent attitudes to aspects of objects according to their appealingness. Furthermore, the OCC model also differentiates the time points (past, now and future) and viewpoints (agent itself or others) of events. The results of the OCC model are usually multiple emotions given one event. Later on, Ortony [100] simplifies the twenty-two distinct emotion types to six positive emotions and six negative emotions.

The **affect dynamics** simulates changes of affective states according to input emotions in interactions. Two dimensional models are proposed to represent affective states: the PA (Pleasure-Arousal) space [72] and the PAD (Pleasure-Arousal-Dominance) space [89]. The PA space adopts pleasure and arousal as two dimensions. The pleasure/unpleasure dimension represents the valence of human affects, while the arousal/unarousal dimension represents the physiological excitedness of affects. The PAD space is a three-dimensional space.

Figure 2.4: The cognitive structure of OCC structure [118]

The additional dominance/submissive dimension represents the activeness of affects, which can differentiate emotions such as “anger” and “fear”. Starting from the ALMA model [45] and other contemporary work, researchers begin to use the PAD space to integrate emotions, moods and personalities together. All of these affects can be mapped into the PAD space. Besides moods that are directly simulated in the PAD space, the mappings of the 22 OCC emotions to the PAD space are provided by ALMA [45], and the mapping matrix of the five factors of personality is provided by [89].

The **affect response** enables social companions to appropriately react to user inputs or environmental changes. Different with traditional decision-making methods in artificial intelligence, the affect response should take into account affective states of social companions. The affect coping as a kind of affect response usually deals with negative affective states (such as “anxiety”, “anger”, “distress”, etc). Psychologists inform us that human tends to keep positive affects while avoiding negative affects [63]. The long-term negative moods could damage both the psychological and physiological health of human. The coping methods contain approaching and avoiding methods. The approaching methods make

social companions solve problems by active planning or asking for others' help, while the avoiding methods make social companions escape from the negative circumstances [52, 43].

Our work in Chapter 4 covers all the above components but mainly focuses on the affect dynamics component, which positions in the medium of affective systems connecting inputs and outputs. The affect dynamics is the key component that makes social companions emotionally autonomous by adapting affective states with interactions. Otherwise, users will have no feeling that companions have evolved in interactions with them and further find they are not human-like and believable.

2.2.3 Computational Models

Inspired by theories in psychology and neuroscience, many computational models have been developed for affect dynamics in companion-user interactions. These models always consider emotions, moods and personalities.

Some affect dynamics models represent affects as discrete states. For instance, fuzzy models usually construct fuzzy affects with all possible affective states. Orozco et al. [99] model personality as a ten-dimensional fuzzy vector, each dimension of which represents one scale of the Minnesota Multiphasic Personality Inventory [124]. Emotions and moods are also modeled as fuzzy vectors with dimensions of all possible states. Instead of representing affects as vectors of discrete values, non-fuzzy methods attempt to model affects with a few key characteristics or dimensions. Egges et al. [35] gives a general model that represents affects in spaces with arbitrary dimensions. However, the space dimensions and the transferring relations between spaces are not specified. Dias and Paiva [31] model emotions with six dimensions (type, valence, target, cause, intensity and time-stamp), and model the valence and mood with one dimension respectively. In addition, personality is modeled as emotional thresholds and decay rates. Becker et al. [8] build a space with valence of emotion, valence of mood and boredom as the three dimensions. However, more work [45, 111, 70] prefer to model affects in the PAD space proposed by Mehrabian [89]

introduced in section 2.2.2. Compared with the Becker's model, for example, the PAD space has validated plausibility in psychology. In the PAD space, affects (including emotions, moods and personalities) are represented as vectors in three dimensions. Our work in Chapter 4 adopts the PAD space too.

Many mechanisms of affect dynamics have been proposed. Becker et al. [8] proposed a mass-spring model that uses two springs to model the dynamics of the valences of emotion and mood, and the emotion valence can also influence the mood valence. Moreover, coefficients of the two springs represent personality. The valences of emotion and mood are coordinates of a reference point in the dimensional space. A reset force will be generated when the reference point is away from the origin. Finally, the emotion valence, mood valence and personality determine the output categories of emotions. This work inspires our work in Chapter 4 that personality should directly influence dynamics of emotions and moods. Gebhard [45] proposes the ALMA model (A Layered Model of Affect) that use a pull and push method to simulate smooth and steady changes in mood. A virtual emotion center (VEC) is defined as the average of all active emotions. Mood is attracted by the VEC if the mood position is between the origin of the PAD space and the VEC position, and is pushed away if it is beyond the VEC position. The strength of the push or pull force is determined by distance of the VEC to the origin. Moreover, the work requires the mood to start from and finally decay to the personality position. Kasap et al. [70] follows the ALMA model but takes relationship of social companions with users into consideration. The relationship is modeled by the friendliness and dominance and updated after each interaction. It changes the starting position of moods.

Some work adopt mechanisms of fuzzy affect dynamics. Orozco's model [99] update affective states in a complicated manner. The dynamics of emotion, mood and personality are mutually influenced by each other. The affect dynamics has several steps. Firstly, the emotion is updated by the personality, mood history and input events by predefined fuzzy rules. Then, the mood is updated by the personality, emotion history and input events. Finally, the emotion and mood are regulated by the personality and input events. Another

Figure 2.5: Case study from Orozco et al. [99]. The left and right figures show the reactions of a hysteric and a depressive virtual human for a very negative event respectively

example is the work of Van der Heide and Trivino [129], which designs a fuzzy finite state machine (FFSM) to memorize current fuzzy affective states. By designing sets of Takagi-Sugeno-Kang (TSK) fuzzy rules, the current fuzzy states are updated by the input fuzzy states. The activation is obtained by defuzzifying the states in FFSM. However, to simulate certain personality, the design of fuzzy rules seems non-intuitive and time-consuming. Moreover, it is difficult to prove the plausibility of these fuzzy rules in complicated interaction scenarios.

Besides fuzzy models, Bayesian Belief Networks (BBN) are also used to simulate uncertainty of human behaviors. Kshirsagar [71] proposes a layered personality model which represents mood in different states. The state transition of mood is determined by its current state, personality, interaction history and results of event appraisals. A drawback of this approach is that they have to manually set reasonable transition probabilities of moods for every task.

Although there lacks well-accepted benchmark to evaluate computational models of affect dynamics, prior studies usually use two types of validation methods, case studies and user studies. We deliberate the two validations below.

Figure 2.6: Case study from Becker et al. [8]. The virtual human gets angry and leaves the screen if the user keeps insulting him

Case Study Case studies [99, 8, 71, 57] run experiments to test if models can simulate reasonable affect dynamics given a few event inputs in a well-designed scenario. For example, how moods of a social companion with a particular personality will change given a positive event followed by a negative event. The outputs could be curves representing dynamics of affective states or sequences of behaviors changing with time. The tendency of affect dynamics in case studies should follow human commonsense.

For instance, Orozco et al. [99] design a case study to evaluate their fuzzy model. The affect dynamics of a hysteric virtual human for a very negative event are compared with those of a depressive virtual human in the same situation, as shown in Figure 2.5. Becker et al. [8] test their affect dynamics model in a scenario that a virtual human is offended by users using insulting words again and again, as shown in Figure 2.6. The emotion of the virtual human changes from “anger” to “annoyance”. The bad moods of the virtual human become increasingly intensive, and finally result in bad words and angry faces. If users stop talking, the moods of the virtual human will return back to calm. The restoring time can be shortened or extended if users continue to compliment or insult the virtual human.

User Study User studies [47, 115, 68] are designed with a few hypotheses. Participants are required to fill a questionnaire after they have completed some tasks. The answers are analyzed statistically to evaluate the proposed hypotheses.

We discuss two example usages of user studies in previous work. Gebhard and Kipp [47] design a user study to evaluate the plausibility of emotions and moods generated by the ALMA model [45]. Given some conversational scenarios, participants are required to evaluate the affect plausibility for 24 types of emotions and 8 types of moods using a discrete ranking scale. Results show that most emotions and moods generated by the ALMA model have significant plausibility than neutral level, except for the emotion “fear” and “hate” as well as the mood “dependent” and “anxious”. Feng and Tan [115] also design a user study to compare their CRAA emotion appraisal model with three other models. Participants are required to play with non-player characters (NPC) integrated with the CRAA appraisal models in a first person shooting game, Unreal Tournament 2004 (UT2004). The NPCs express their emotions by words. In the first experiment, given a description of typical situations in UT2004, participants are required to appraise the plausibility of the appraisal variables and emotions generated by the CRAA model and the other three models. The results suggest that the accuracy of the CRAA model significantly outperform the others. The second experiment requires participants to play with the NPCs with or without the CRAA model in UT2004. After games, participants are required to evaluate the accuracy of emotion appraisal and how interesting to play with the NPCs. The results indicate that the CRAA model enhances the attractiveness and playability of the game.

2.3 Episodic Memory

Episodic memory is an indispensable module towards a persistent engaging social companion-user interaction. In this section, we first introduce theories of human memory from the viewpoint of psychology and neuroscience. Then we present main trends of current researches in computational episodic memory models.

2.3.1 Background in Psychology and Neuroscience

The researches on human memory from psychology and neuroscience start from the last century. The biological nature of memory is too complicated to be fully understood yet. As shown in Table 2.3 (taken from [138]), previous studies separate human memory into multiple subsystems. Based on the duality of retention, human memory, as originally proposed by Hebb [59], can be divided into short-term memory (STM) and long-term memory (LTM). STM, or known as working memory (WM), serves as a central executive component with limited capacity processing multi-modal information from perceptions to enable filtering and instant retrieval. LTM stores memory records for a much longer duration ranging from hours to whole lifetime. Only partial information, with repeated exposure or deliberated attention, has the opportunity to be transferred from STM to LTM. On the other hand, as shown in Figure 2.7, LTM can be further separated into declarative and non-declarative memory based on distinctive contents and functionalities. Non-declarative memory (mainly procedural memory) concerns the non-conscious and implicit process of grasping skills or habits, and stores information of how to perform certain procedures. On the contrary, declarative memory stores facts or knowledge that can be consciously recalled. More specifically, declarative memory is constituted by semantic memory and episodic memory. Semantic memory concerns general knowledge and rules without a context, while episodic memory collects past personal experience with specific time, location and context [126]. The relations of episodic memory and semantic memory are known as bi-directional. On one hand, episodes are constituted by temporal sequences of events with tuples of semantic units. On the other hand, semantic knowledge is learned from accumulated episodes with common features. In addition, episodic memory allows people to travel back in time track and experience similar but weaker feelings from self-perspective. This feature of introspection differentiates human with animals that have episodic-like memory though [2].

Anatomical evidence suggests that hippocampus, as part of the medial temporal lobe,

Table 2.3: Definitions for various memory subsystems

Term	Definition
Memory	The capacity to use previous experience to inform subsequent behavior
Long-term memory (LTM)	Temporally indeterminate, independent of specific task demands
Short-term memory (STM)	A functionally distinct memory structure, finite in capacity and retention period, bounded by context and task demands
Working memory (WM)	A variant of the STM concept providing temporary retention of task-relevant information
Procedural	Non-consciously accessible memory e.g., skilled motor behavior, habits, and stimulus C response conditioning
Declarative	Consciously accessible information
Episodic	Personally experienced event information, spatially and temporally organized
Semantic	Context independent information, facts and concepts

Figure 2.7: Relations of memory subsystems

plays an important role in human episodic memory [2]. Damages in this area lead to anterograde amnesia for new autobiographical information. In addition, the temporal sense in episodic memory is closely related to prefrontal cortex [64]. Patients with prefrontal cortex impairment can learn new episodes, but with confusion on the temporal order of events. The anatomical structure of brain components related to episodic memory is shown in Figure 2.8 (taken from [138]).

Figure 2.8: Anatomical human brain components related to episodic memory

2.3.2 Memory Requirements and Functionalities

Tecuci and Porter [123] propose five requirements for good computational episodic memory (taken from [123]):

- (1) **Accuracy** Memory should return the most relevant records to given cues.
- (2) **Scalability** Memory should be able to accommodate a large number of episodes without a significant decrease in performance
- (3) **Efficiency** Memory should provide efficient storage and recall
- (4) **Content addressability** Memory items should be addressable by their content
- (5) **Flexible matching** Memory should robust recall for noisy or partial cues.

Table 2.4: General functionalities of computational memory models [98]

Functionality	Task	Definition
Encoding	Encoding initiation	When the encoding of an episode should start
	Episode determination	What information should be encoded
	Feature selection	What features should be selected for retrieval
Storage	Episode structure	How the episode should be stored
	Episode dynamics	How stored episodes should adapt over time
Retrieval	Retrieval initiation	When the retrieval should start
	Cue determination	What features should the retrieval cues contain
	Retrieval	How to retrieve relevant episodes given cues
	Retrieval representation	How the retrieved episodes should be represented to agents or users
	Retrieval meta-data	What meta-information should be used

The functionalities of episodes have been fully discussed in psychology and neuroscience [126]. Nuxoll and Laird [98] first systemically introduced these functionalities, including encoding, storage and retrieval, into computational models of episodic memory. The encoding extracts important features from current situations and represents them in appropriate data structures. The storage finds effective algorithms to store and organize these episodes. Given retrieval cues, the most matched episodes should be retrieved by relevances to the cues and used to solve problems. Each functionality includes several subtasks as shown in Table 2.4 [98].

2.3.3 Computational Episodic Memory Models

Various computational models for episodic memory have been proposed in recent years. From the viewpoint of application, we categorize these work to two types: (1) memory models for intelligent agents, and (2) memory models for social companions. The two categories of models share fundamental goals and requirements that they should provide efficient storage and accurate retrieval for agents or companions. However, the main difference is that how to make use of retrieved episodes for specific applications.

The episodic memory models for intelligent agents are mainly aimed at improving agent

performance by learning from past experience. Nuxoll and Laird [98] design an episodic memory model for a general cognitive architecture, SOAR [96]. The model is evaluated in a game TankSoar, as shown in Figure 2.9 (a), by improving scores of a tank agent. The tank agent can get scores by destroying other tanks while avoiding being destroyed. In this game, the episodic memory supports three cognitive abilities: virtual sensor, action selection and strategy learning. However, the memory model retrieves episodes only based on the number of shared elements which may result in low noise-resistance. In addition, the flat memory structure only supports serial searching which is slower than hierarchical searching. Ho et.al [61] design an episodic memory model that helps agents survive longer in a complex and dynamic virtual environment, as shown in Figure 2.9 (b). The memory model endows the agents with abilities of making decisions, finding resources and planning paths. However, the loosely organized memory structure (the whole memory is not separated into episodes) may result in low retrieval efficiency. Wang and Tan [135] propose an EM-ART model, a hierarchical neural network for episodic memory. The model is integrated with agents in a first-person shooting game, Unreal Tournament 2004 (UT2004). The episodic memory helps the agents select powerful weapons and make correct decisions in different contexts. Our work is inspired by the hierarchical structure of the EM-ART model which will be deliberated in Chapter 5.

The episodic memory models for social companions mainly focus on improving user engagement by recalling related past experiences during interactions. The companions with episodic memory give users impressions that they are growing during interactions with awareness of what has happened in the past. Different from models for agents, these work always consider emotional salience and time recency of events or episodes as additional factors for memory retrieval. Dias et.al [30] build companions with long-term episodic memory, FAtiMA, to self-report their past experiences and rationale of actions in an anti-bully scenario FearNot, as shown in Figure 2.10 (a). The most significant events in terms of emotional saliences are stored and reported to users. However, this work is task-oriented by only remembering events with high emotion impacts or task importance thus it can only

Figure 2.9: Example testing environments for episodic memory models designed for intelligent agents. (a) is the TankSoar game used in Nuxoll and Laird [98], (b) is the dynamic environment used in Ho et.al [61]

work for task-related cues. In addition, this memory model seems not content-addressable. Brom et.al [14] attempt to realize story-telling by creating a full episodic memory in a simple RPG game. The memory enables non-player agents (NPA) to answer users' questions based on their past experiences. In this work, episodes encode working memory elements such as tasks, phantoms (objects with states) and time-pointers (time, goals and rewards). A tree-like structure is adopted to store episodes with hashed phantoms and time-pointers indexed by days. The memory retrieves relevant tasks in different layer of the tree structure to answer questions like "what was the actor doing from t_1 to t_2 ". The tasks are retrieved according to cues with phantoms and time-pointers. Useless episodes are forgotten based on their importance, emotional saliences and task complexity levels. However, this work is task-oriented because it is designed for the specific task of reconstructing events in some time spans. Thus it may not be suitable for general episode retrieval without a certain goal. Dodd [33] design an episodic memory for ISAC humanoid robot. In this work, episodes are stored as collections of semantic units of working memory. The retrieval relevances of episodes are related to the emotional saliences, usage frequencies and occurrence frequencies of these semantic units. The memory model stores episodes with flat structure and will

(a) FearNot

(b) Eva

Figure 2.10: Example testing environments for episodic memory models designed for social companions. (a) is the FearNot story-telling game used in Dodd [33], (b) is the robot tutor Eva used in Kasap et.al. [70]

forget those which are rarely used. The memory model is tested by making the ISAC robot to report fire emergency. However, the ISAC model fails to consider the order of events in episode retrieval. Thus it is not robust to differentiate episodes that are extremely similar. Kasap et.al. [68] develop a long-term episodic memory to provide a robotic tutor Eva, as shown in Figure 2.10 (b), with awareness of past interaction history. The memory is also linked with an affective system to evaluate relations with users. The robotic tutor can recall previous performance of students on some questions during interactions. However, the database-like storage and retrieval are too shallow to provide powerful episodic memory functionalities.

Some studies focus on designing episodic memory for free-dialog based social companion interactions. Different with the above models that work in small domains with predefined tasks and limited states, memory models for free-dialog based interactions have large scales of attributes due to tremendous words in natural language and more complicated events and episodes to memorize. To realize memory functionalities, some studies [1, 24] collect user personal information by asking certain questions in appropriate time. Nevertheless, most studies [18, 104, 116, 137] generate responses and make simple reasonings by extracting Resource Description Framework (RDF) knowledge from user speech.

The adopted knowledge extraction methods are good at managing information in database and integrating online knowledge resources such as ConceptNet* or DBpedia†. Sieber and Krenn [116] design an episodic memory model to enhance dialog ability of a virtual companion on topics of biography of known artists. Towards open dialog interactions without focusing on any specific plan, the episodic memory is designed to be compatible to RDF knowledge and has access to online RDF database. The memory-enabled dialog can adapt to user interests and preferences. Episodes contain common attributes such as time, actors and episode ids, and are stored in SeRQL database. They are retrieved either by the recency or matched patterns in user speech, or by SeROL retrieval for the most relevant events or episodes. However, the RDF-based memory has no consideration of the order of events in episodes. Another example is the work of Wilks et.al [137], which develops a social companion to “assist, amuse, entertain and gain the trust of the user” by communicating with “coherent life narrative”. The social companion discusses with users about their photographs. The topics cover the time, locations and people in photos. Knowledge is extracted from user speech with a relation extraction method and are stored in a SQL database. In addition, the social companion can do simple reasoning based on the learnt knowledge and maintain user moods by carefully select verbal responses. However, these extraction-based memory models have following problems: (1) The methods have strict format requirements for user speech inputs. Thus, they always fail for incomplete or informal sentences within contexts, which are frequent in spoken languages. (2) The methods have few consideration of synonymy and polysemy in natural language. (3) Most methods are domain-dependent. For example, they need to predefine manually the domain ontologies and rules for extraction.

*<http://conceptnet5.media.mit.edu/>

†<http://wiki.dbpedia.org/>

Chapter 3

System Architecture for Social Companions

3.1 Our Social Companions

Social companions, getting increasing attention in recent years, can interact with people as capable assistants, knowledgeable tutors, considerate doctors, smart buddies and loyal friends. A social companion has to integrate many modules to achieve natural interactions.

There are some existing open-source platforms and tools for the design of interactive social agents. The Virtual Human toolkit [58] provides a comprehensive framework with multiple capabilities that assist users to rapidly create new virtual characters and systems. The toolkit contains both prototypes of virtual characters and many modules such as the speech recognition, natural language understanding, behavior understanding and behavior generation, etc. SAIBA [131] provides a general framework for agent behavior design with several “standard” script languages. These languages include the Function Markup Language (FML) for planning, the Behavior Markup Language (BML) for behavior generation and the Emotion Markup Language (emotionML) for emotion generation. Social Signal Interpretation (SSI) [134] provides tools to record, analyze and recognize signals, such as speeches, gestures and facial expressions, from environment. IntraFace [27] provides tools

for facial feature detection and tracking as well as facial expression analysis.

In this chapter, we propose a general system architecture for social companions with all necessary components, modules and their connections to achieve certain degree of autonomy. Compared with the above open-source platforms, the proposed architecture is more suitable for our applications and easier to integrate new research outcomes of our team. In addition, few open-source platform has integrated some new features such as episodic memory. We have our humanoid robot Nadine and virtual human Sophie share the same platform so that many modules are available for both.

Figure 3.1: The Nadine social robot

3.1.1 Our Social Robot

Nadine is a realistic female humanoid robot with her physical body developed by Kokoro Company, Ltd*. On the appearance level, Nadine has realistic artificial skin and believable appearance. On the behavior level, a total of 27 degrees of freedom (DoFs) for body and face enables natural facial expressions and body movements. Figure 3.1 displays the Nadine social robot in our demo settings.

*www.kokoro-dreams.co.jp/english/

3.1.2 Our Virtual Human

Figure 3.2: The Sophie virtual human

As a counterpart of Nadine, Sophie is an appealing virtual human developed by the Institute for Media Innovation (IMI)[†]. The appearance and animations of Sophie are developed in 3DS MAX[‡] and the behavior mechanism of Sophie are developed in Unity3D[§]. Figure 3.2 displays the graphic effects of Sophie.

3.1.3 Distinctions

On the software level, both our social robot and virtual human share the same platform, except that the virtual human has more DoFs and is controlled virtually in the Unity3D, while the robot is controlled by motors and air compressor. The platform sharing benefits our development that the new modules for the robot can simultaneously improve the virtual human and vice versa.

On the interaction level, the physical embodiment of the social robot provides more believable user experience than the virtual human. Users are more easily attracted by the

[†]www.imi.ntu.edu.sg

[‡]<http://www.autodesk.com/products/3ds-max/overview>

[§]<http://unity3d.com/>

facial expressions and gestures of the robot than the same designs on the virtual human. The physical touches and eye contacts make users feel realistic.

Compared to robots, virtual humans have the advantages of low cost and high portability. Virtual humans can be cheaply reproduced or adapted while robots cannot. In addition, virtual humans embedded in smartphones or tablets can interact with users no matter when and where, while robots have a limited moving scope.

A promising vision in future is the cooperation of social robots and virtual humans. Users interact with a social robot at home or office while interact with a virtual version of the robot, which shares the same memory and cognition, on tablets on traveling. Meanwhile, users can have remote control or can be represented autonomously by robots via telepresence.

3.2 Proposed System Architecture

Figure 3.3: The system architecture of our social companion

The proposed system architecture endows our companions with certain degree of autonomy. The autonomy is displayed by their strong self-consciousness. Instead of following

predefined scripts or instructive controls by human beings, our social companions make decisions according to their own demands, values, moods and personalities. They can collect environment information from video and audio sensors, shift their intention to dynamic contexts, analyze semantics behind signals, do reasoning and make sensible responses, have emotional reactions and learn from memorized past experiences.

In addition, our social companions own some social abilities and communication skills. They interact people with multiple languages in an elegant way, keep valuable politeness and empathy, and always have a good sense of humor. The occasional displays of their emotions and current moods by facial expressions, gestures and speech tones make them realistic and believable for human beings.

Figure 3.3 shows system modules of our social companions. All the modules can be grouped into three main components: (1) Perception, (2) Internal processing and (3) Action.

At the first place, the perception component collects input signals from environment, in the forms of video or audio, and transforms them into semantic information. The environmental inputs are caught by a Kinect 2 and a microphone. This component contains many modules, such as the recognition of user identity by user faces or voice, recognition of body or hand gestures, user localization by skeletons or sounds and speech recognition.

The internal processing component simulates human-like cognition by analyzing semantic inputs from perception to generate appropriate responses. This component considers the user tracking, behavior selection (or decision making), planner, affective system and episodic memory.

Finally, the action component sends commands to the hardware of our social companions to produce verbal and non-verbal behaviors. The verbal behaviors are generated by a dialog manager, which includes a text generator and text-to-speech module. Meanwhile, the non-verbal behaviors, such as gestures, facial expressions, head and gaze movements, are performed by a companion controller.

To realize flexible integration and communication among all modules, an Integrated Interaction Platform (I2P) is developed. I2P is an independent platform framework with

Table 3.1: Communication of modules of social companion

Modules	Category	Input	Output
Face recognition	Perception	image	user identity, gender
Voice recognition	Perception	audio	user identity, gender
Action recognition	Perception	image, depth, skeleton	user activity
Object recognition	Perception	image, depth	object
Gesture recognition	Perception	skeleton, depth	user gesture
Facial emotion analysis	Perception	image	user emotion
Voice emotion analysis	Perception	audio	user emotion
Skeleton localization	Perception	skeleton	user location
Audio localization	Perception	audio	user location
Speech recognition	Perception	audio	user speech
Behavior-Tree Planner	Processing	perception & internal info	tasks and states
Behavior manager	Processing	tasks and states	behaviors
Dialog manager	Processing	tasks and states	dialog and tone
Episodic memory	Processing	user identity, gender, speech	retrieved past events or utterances
Affective system	Processing	user speech, emotions	self affect states
Nadine controller	Action	behaviors and dialog	actions

modularity and low network latency. The data exchanging and event publishing among modules utilize the remote procedure calls (RPCs) provided by the Thrift[¶] framework on client/server architecture. Different programming languages, C++, Java and Python, can be used alternatively for system development. The communication and exchange among modules are displayed by Table 3.1.

3.3 Modules

Among all the modules in the system architecture, the affective system, episodic memory and dialog manager are developed by myself. For the other modules, we adopted either open-source software or work from other team members with necessary adaptations. We

[¶]<http://thrift.apache.org/>

introduce some modules in brief below.

3.3.1 Perception

Skeleton Capturing

The Kinect 2 is adopted to track skeletons for six users concurrently and to detect 25 skeletal joints per frame. To avoid disturbance of irrelevant people, we use a depth control with 5 meters. The closer the distance of a user is, the easier to get attention of our social companions.

Sound Localization

Besides reacting to video inputs, the social companions also focus on users who are speaking. The sound localization from the Kinect 2 is used to get world coordinates of the speakers.

Face Recognition

To interact with users, the social companions need to identify them. Our system adopts a face recognition module [107] which consists of three subsystems: face detection, face alignment and face identification. The main idea is to use the local binary pattern (LBP) texture features of user face images to detect user identities. Figure 3.4 displays a diagram of the face recognition module.

To train new users, we store the LBP features extracted from user faces as templates. Their identities can be recognized by comparing features of new face inputs with the stored templates to find the closest match.

Figure 3.4: The diagram of face recognition system [107]

Voice Recognition

An alternative way to recognize user identities is from their voices. The Microsoft speaker recognition ^{||} is integrated with our companions. The results of face recognition and voice recognition are analyzed synthetically.

Gesture Recognition

Two methods are developed for user gesture recognition. One method [139] recognizes user upper body gestures by combining information from a Kinect and a CyberGlove^{**}. As displayed in Figure 3.5, the head and arm postures captured by the Kinect and the hand postures captured by the CyberGlove are combined together to classify 12 upper body gestures.

The other method [79] is less user-intrusive. It recognizes user gestures from image inputs by a majority vote of all pixels for classification.

^{||} www.microsoft.com/cognitive-services/en-us/speaker-recognition-api

^{**} www.cyberglovesystems.com

Figure 3.5: The diagram of upper body gesture recognition [139]

Speech Recognition

A python implementation^{††} of Google speech recognition API is adopted to recognize user utterances. The speech recognition module catches user speech inputs in multiple languages and generates the corresponding texts.

3.3.2 Internal Processing

Behavior Selection

A behavior selection module with an activation-based competition mechanism is developed. The module can distribute attentions of our social companions to multiple users according to perception inputs. The activations of behaviors are updated by multiple environmental inputs, such as the social distance, user speech and behaviors. All activated behaviors participate the competition while only the winner is executed.

Planner

As shown in Figure 3.6, an independent behavior-tree-based planner [80] is implemented to control the workflow of social companions. The independence provides convenience for adding or deleting tasks. The behavior tree, constituted by leaf nodes and kinds of decorators to control, can facilitate the design of logics of actions to achieve complex goals.

^{††}<https://pypi.python.org/pypi/SpeechRecognition/>

Figure 3.6: The implemented behavior-tree planner

The leaf nodes contains *Action* nodes and *Condition* nodes. The *Action* nodes execute primitive actions, and the *Condition* nodes checks if desired states are satisfied for certain tasks.

Four decorators, *Loop*, *Selection*, *Sequence* and *Parallel*, are used in this work to control the logic flow. A *Loop* decorator controls an iterative execution of its subtrees. A *Selection* decorator executes the first successful child node and returns a failure unless every child node fails. A *Sequence* decorator executes its subtrees sequentially until it meets the first failed child node. A *Parallel* decorator executes its subtrees concurrently and returns a success if all child nodes succeed.

Affective System

An affective system simulates human affects including emotions, moods and personalities, from the short-term and superficial to the long-term and deep nature of affects. User speech inputs are first appraised by an emotion classification module SenticNet [17] and then transferred to six kinds of OCC emotions [101]. The six emotions are “Happy-for”, “Pity” (Appraisals from companions to users. For example, the companion is happy for a user if he says “I am happy”), “Gratitude”, “Anger” (Appraisals from users to companions. For example, the companion feels angry for a user if he says “You are ugly”), “Joy”, “Distress” (Appraisals for objective events. For example, the companion feels joy if a user says “It is sunny today”). These emotion stimuli will be processed by a personality-characterized mood dynamics (PCMD) module [140] to update the affective states of companions. The proposed PCMD module will be deliberated in Chapter 4.

Episodic Memory

An episodic memory module encodes companions’ behaviors with contexts into episodes. The contexts contain the objects, subjects, actions, time, locations and user utterances of events. If requested by the dialog manager, the memory module can retrieve relevant past events or episodes given user speech cues and current contexts. The proposed mixed-correlation-analysis based episodic memory (MCAEM) module will be described in Chapter 5.

3.3.3 Action

Dialog Manager

The dialog manager makes appropriate responses for user speech inputs. As shown in Figure 3.7, the process is below. Firstly, the dialog manager tries to retrieve relevant events or episodes by the episodic memory modules and to paraphrase sentences in the retrieved events with necessary transformations of pronouns and time based on the current contexts.

Figure 3.7: The diagram of dialog manager

This process is deliberated in Chapter 6. Then if no matched memory is found or the questions are not for deliberate retrieval of past experience, the manager will get further answers from a local AIML 2.0 chatbot^{††}, which is an open source Java implementation of the AIML 2.0 draft specification. The database of the AIML chatbot is adapted to our current application. The chatbot can respond to natural language inputs by pattern matching (partial matching followed by exact matching) with certain abilities of knowledge learning. Finally, if we can't get satisfied answers from chatbot, the manager will search online for question answering or news searching, or ask users for correct answers.

Companion Controller

A companion controller synchronizes the animations, speech and gaze of social companions. The controller contains the lip synchronization, blending mechanisms, idle behaviors, signal attention and online motion generation.

Offline Animation Design Towards natural companion-user interactions, some human-like gestures and facial expressions are designed for Nadine and Sophie. The animations of Nadine are designed by the Kokoros WTEN software in a XML format, and those of

^{††}www.alicebot.org

Figure 3.8: Example gestures of Nadine

Sophie are designed by the 3DS MAX and are exported to Unity3D. The example gestures of Nadine and Sophie are shown in Figure 3.8 and 3.9.

Online Motion Generation The companion needs to generate online motions, such as shaking hands, tracking and pointing to users. The algorithm described in [141] is adopted to generate head-arm motions. The head motions are generated by a virtual plane approach, and the coordinated dual-arm motions are generated by a quadratic program (QP)-based method.

Figure 3.9: Example gestures of Sophie

Gazing A real-time gaze animation algorithm [7] is adopted on our companions. To keep gazing at users, this algorithm provides social companions an analytical solution to coordinate movements of the neck and eyeballs.

Lip Synchronization To perform natural speech, we adopt the CereProc Cerevoice Text-to-Speech (TTS)* to transfer texts to audio files in multiple languages and emotional tones. The audio files are used to generate phonemes by a tool of lip synchronization. The audio files and lip animations are executed simultaneously.

*<https://www.cereproc.com>

Chapter 4

PCMD: Personality-Characterized Mood Dynamics

To automatically generate appropriate emotional behavior patterns to display particular personalities, we propose in this chapter to use the mood to connect behavior patterns and personalities (here it mainly refers to temperaments) of social companions. Thus given well-designed rules of mood-driven behavior selections, the problem can be solved by personalizing the mood dynamics. A personality-characterized mood dynamics (PCMD) model is proposed to make behavior patterns of companions consistent to convey their personalities and flexible to respond to various situations.

4.1 Introduction

Towards natural companion-user interactions, it is effective to design an abundant repository of human-like behaviors, including facial expressions, gestures and dialogs. In this chapter, the problem to be considered is: given a well-designed repository of emotional behaviors, how to automatically organize these behaviors in a specific way to show certain personalities. The proposed approach is to take the mood as an intermediate concept and solve the problem by making the dynamics of moods exhibit certain personality patterns.

Figure 4.1: The rationale of the PCMD model for social companion

This is because mood, as mentioned in section 2.2.1, is a long-lived affect states that influences people on their behavior patterns and strategies. The rules for behavior selections by current mood and emotion can be heuristically designed. For example, pleasant moods and emotions can lead social companion to smiles, cheerful gestures and joyful speech tones. As illustrated by Figure 4.1, the problem of how to get the personalized behavior patterns can be transferred to the problem of how to make the dynamics of moods show particular personalities. Therefore, we propose a personality-characterized mood dynamics (PCMD) model, which makes the overall mood tendency align with the given personality inputs by solving an offline optimization problem.

The inputs to the PCMD model are a user-specified personality in the form of Big-five [50] or PAD factors [89] and the probability distribution over all available OCC emotions. The outputs are groups of appropriate weights for emotions that makes the overall mood approach the given personality.

4.2 Related Work

The behaviors of social companions are deeply influenced by their affects. Early studies, such as Velásquez [130] and El-Nasr et.al [37], take efforts to build affective systems to influence the decision-making. Marsella and Gratch [84] design a bunch of emotion appraisal and coping strategies to model the affective reactions. However, these studies fail to model the concept of personality explicitly, which makes the behaviors can't be characterized by the personalities of the agents.

Some studies personalize social companions by heuristically designing the verbal or non-verbal behaviors for specific personalities. For verbal behaviors, Mairesse and Walker [83] develop a linguistic style generator PERSONTAGE that mimics the talking styles of users by parameter adjusting. For non-verbal behaviors, Gebhard et.al [46] develop two virtual characters as job interviewers with different personalities, understanding or demanding. However, these studies make direct connection between personalities and behavior patterns, which makes the behavior patterns consistent but inflexible for situations. In addition, the manual designs of the behavior patterns can not work for arbitrary personalities.

A variety of studies focus on the dynamics of the emotions and moods under the influence of the personalities to select appropriate behaviors for situations. For example, Santos et.al [111] model the personality, emotion and mood in a negotiation scenario among group of participants. Kasap et al. [70] design a virtual teacher with emotion and memory system. This paper is inspired by the ALMA model [45], that combines the emotions, mood and personality together in the PAD space [89]. In this model, the personality determines the start position and the decay direction of the mood and influences the emotion intensities in the appraisal process. In their later work [47], the emotion intensities are also influenced by the current mood. Unfortunately, these previous studies have difficulty in generating consistent behavior patterns to convey the intended personalities, because they exclude the consideration of making the tendency of the mood consistent to the personality after sufficient interactions.

This work adopts the ALMA model as the starting point because it models the personality explicitly, and the relation among emotions, moods and personalities can be easily handled in mathematical way (by dot product). We propose an optimization method to automatically determine the emotion weights such that the tendency of the mood aligns with the personality after sufficient interactions. Thus the overall mood-driven behavior patterns can convey the personalities of social companions.

4.3 Preliminaries

In this section, we first introduce some psychology concepts and notations to be used in later sections, then we describe the mood dynamics of the ALMA model [45] as the starting point of our discussion.

4.3.1 Psychology Concepts and the PAD space

Human affects can be classified into emotions, moods and personalities. Emotions are short-lived affects that are usually bound to particular objects or situations [44, 26]. Moods is longer-lived affect and has great influence on the long-term cognitive strategies [93]. Personality is defined as the predisposition of affect [12].

The PAD (Pleasure-Arousal-Dominance) space [89] is adopted to represent the emotions, mood and personality. An emotion can be represented as a moving point (vector) starting from the origin. The length of the vector represents the emotion intensity and the direction represents the emotion type [106]. Table 4.1 [45] provides the mapping from the 22 OCC emotions [101] to the PAD space. Mood and personality are also represented as a moving point (vector) in the PAD space. Personality is modeled by the “Big Five” traits [50], which can be mapped into the PAD space by the mapping given in Table 4.2 [89].

Table 4.1: Mapping from 22 emotions to PAD [45]

Emotion	P	A	D	Mood type
Hope	0.2	0.2	-0.1	Dependent
Gratitude	0.4	0.2	-0.3	Dependent
Admiration	0.5	0.3	-0.2	Dependent
Gratification	0.6	0.5	0.4	Exuberant
HappyFor	0.4	0.2	0.2	Exuberant
Joy	0.4	0.2	0.1	Exuberant
Love	0.3	0.1	0.2	Exuberant
Pride	0.4	0.3	0.3	Exuberant
Relief	0.2	-0.3	0.4	Relaxed
Satisfaction	0.3	-0.2	0.4	Relaxed
Gloating	0.3	-0.3	-0.1	Docile
Remorse	-0.3	0.1	-0.6	Anxious
Disappointment	-0.3	0.1	-0.4	Anxious
Fear	-0.64	0.60	-0.43	Anxious
Shame	-0.3	0.1	-0.6	Anxious
Resentment	-0.2	-0.3	-0.2	Bored
Fears-Confirmed	-0.5	-0.3	-0.7	Bored
Pity	-0.4	-0.2	-0.5	Bored
Distress	-0.4	-0.2	-0.5	Bored
Anger	-0.51	0.59	0.25	Hostile
Hate	-0.6	0.6	0.3	Hostile
Reproach	-0.3	-0.1	0.4	Disdainful

4.3.2 Notation Declaration

For the ease of discussion, we introduce some notations as listed in Table 4.3 and Table 4.4. The boldfaced letters represent column vectors or matrix and other letters represent scalar numbers. Thus the emotion type corresponds to the emotion index and the direction of the emotion with index i , $\mathbf{D}_i (i \in \Pi)$, comes from the PAD triple values given in Table 4.1. For example, the emotion “Admiration” corresponds to index 1 and has the direction $\mathbf{D}_1 =$

Table 4.2: Mapping from FFM to PAD [89].

	O	C	E	A	N
Pleasure	0	0	0.21	0.59	0.19
Arousal	0.15	0	0	0.30	-0.57
Dominance	0.25	0.17	0.60	-0.32	0

$(0.5, 0.3, -0.2)^T$. All the emotion directions are identical for every companion, except that the available emotions and their probability distributions may be different for applications.

Table 4.3: Notations

Notation	Meaning
$\mathbf{D}$	The list of emotion directions based on Table 4.1
Π	The list of emotion indices based on Table 4.1
S	The total number of emotion index
I_i	The intensity of the emotion of index $i, i \in \Pi$
$\mathbf{E}_i$	The emotion i position in the PAD space
$\mathbf{M}$	The mood position in PAD space
$\mathbf{M}_0$	The initial mood position before interactions
$\mathbf{V}_M$	The mood velocity in PAD space
$\mathbf{F}$	The big five personality factors
$\mathbf{P}$	The personality position in the PAD space
$\mathbf{Q}_{3 \times 5}$	The transformation matrix from Table 4.2
$\{\mu_i\}_{i=1}^S$	The emotion weights for emotions.
α	The attraction parameter from emotion to mood
β	The repulsion parameter from emotion to mood
λ	The frequency of event arrival
γ_M	The mood biasing parameter
γ_P	The personality biasing parameter
δ_e	Emotion decay rate
δ_m	Mood decay rate
Δt	Time step

Table 4.4: Operators

Operator	Meaning
$\langle \mathbf{x}, \mathbf{y} \rangle$	Dot product of two vectors $\mathbf{x}, \mathbf{y}$
$[\mathbf{x}]_a^b$	Normalize each dimension of vector $\mathbf{x}$ to fall in $[a, b]$
$[\mathbf{x}]_a^o$	The normalized vector of vector $\mathbf{x}$
$\mathbf{x}^T$	The transpose of vector $\mathbf{x}$.
$\ \mathbf{x}\ _\infty$	The infinity norm of the vector $\mathbf{x}$.
$\ \mathbf{X}\ _F$	The Frobenius norm of the matrix $\mathbf{X}$.

4.3.3 Mood Dynamics of ALMA

To make computational analysis in section 4.4, we first introduce the mood dynamics process of the ALMA model [45, 47]. According to the ALMA model, the process of mood dynamics is iteratively conducted in several steps outlined below.

- (1) **Initialization:** Various parameters of the model are initialized.
- (2) **Event appraisal:** Whether any new event happens is checked. If yes, the event is appraised and the corresponding emotions are generated.
- (3) **Mood update:** The velocity and the position of the mood are updated.
- (4) **Emotion and mood decay:** Active emotions, if any, decay with time. If currently there is no active emotion, the mood decays with time.
- (5) **Output:** If there are multiple active emotions, the emotion with the maximum intensity value is selected as the dominant emotion. The current dominant emotion and the mood are then passed to the working memory.

The steps from (2) to (4) are iterated. The details of the first 4 steps are described in the following subsections.

Initialization At the beginning of the interaction, the initial intensities of all the emotions are set to zero, i.e., $I_i = 0$ for all $i \in \Pi$. The initial mood velocity is set to the zero vector, $\mathbf{V}_M = \mathbf{0}$. The initial mood position could be anywhere in the PAD space in this work, but is better set to the origin if nothing has happened before (ALMA initializes the mood by the personality to enhance the first impression of the users). The personality $\mathbf{F}$ and the constant parameters $\alpha, \beta, \lambda, \gamma_M, \gamma_P, \delta_e, \delta_m, \Delta_t$ need to be predefined. The personality in the PAD space is obtained by the formula $\mathbf{P} = \mathbf{Q} \times \mathbf{F}$ [88]. The emotion intensities $I_i, i \in \Pi$ are between 0 and 1, each coordinate of vectors in the PAD space is between -1 and 1, and each personality trait of $\mathbf{F}$ is between -1 and 1.

Event appraisal The appraisal of emotions is determined by the event, the current mood and the personality. For a new emotion with index i , we have

$$I_i = [I_{i0} + \gamma_M \langle \mathbf{M}, \mathbf{D}_i \rangle + \gamma_P \langle \mathbf{P}, \mathbf{D}_i \rangle]_0^1 \quad (4.1)$$

where I_{i0} is the emotion intensity by the OCC appraisal [101], $\mathbf{D}_i$ is the direction of the emotion, and γ_M and γ_P are the mood and personality biasing parameters. The intensity is normalized into the interval of $[0, 1]$.

Then the position of the new emotion in the PAD space is

$$\mathbf{E}_i = I_i \mathbf{D}_i, i \in \Pi. \quad (4.2)$$

Mood update After event appraisal, the active emotions make changes on the mood. The mood velocity is computed from the active emotions $\mathbf{E}_i$:

$$\mathbf{V}_M = \sum_i \alpha \mathbf{E}_i. \quad (4.3)$$

Formula (4.3) is adopted from [70], which is a simplified version of the ALMA model [45] with the following assumptions: (1) The active emotions contribute to the mood changes independently; (2) The attraction parameter equals to the repulsion parameter, i.e., $\alpha = \beta$; (3) The three dimensions of the PAD space are independent of each other [89].

The mood position is then updated from the recent one by

$$\mathbf{M}(t + \Delta_t) = [\mathbf{M}(t) + \Delta_t \mathbf{V}_M]_{-1}^1, \quad (4.4)$$

each component of the mood is normalized into the interval of $[-1, 1]$.

Emotion and mood decay In each time step, the active emotions decay at an exponential rate:

$$I_i(t + \Delta_t) = I_i(t) e^{-\delta_e \Delta_t}, i \in \Pi. \quad (4.5)$$

If there is no active emotion, the mood starts to decay in an exponential rate:

$$\mathbf{M}(t + \Delta_t) = \mathbf{M}(t) e^{-\delta_m \Delta_t}. \quad (4.6)$$

4.4 Personality-Characterized Mood Dynamics (PCMD)

4.4.1 Rationale

Many computational affect dynamics models have been proposed in recent years as shown in section 2.2.3. These models formulate the relations of emotions, moods and personalities in kinds of ways following some disciplines from psychology and neuroscience. However, these models have weakness of lacking clear objective functions, which make the evaluation and comparison difficult. Taking the ALMA model [45] for example, it is hard to explain why mood should start from the personality position instead of the origin in the PAD space.

We proposed our motivation in the section 4.1 to generate personalized behavior patterns by making the dynamics of moods exhibit certain personality patterns, so that users can recognize the personalities of companions after several interactions. Differing from prior studies, we propose a clear objective that the overall mood dynamics (namely, the expectation of the mood direction) should approach the personality direction after any sufficient interactions. This objective, which has not been achieved by prior models, can help us to optimize and evaluate the relations of emotions, moods and personalities. Here we emphasize the mood direction (not the mood intensity). This is because the mood direction determines the mood type which further influences behavior patterns.

To achieve the above objective, we assign weights for emotions which can differentiate their effects in driving mood dynamics. This configuration is inspired by our observation that people of distinct personalities have different reactions of moods to the same emotions. For example, calm people have better control of their angers for some irritating events. One possible explanation is that the anger emotion has less influence on moods of people with clam personalities.

One may argue that the biasing effects of mood and personality at event appraisal and

enforcing mood to start from the personality position in the ALMA model could achieve similar effects too. However, the numerical simulation in section 4.4.3 doesn't support this argument because the influence of the biasing effects and the initial position is too indirect to characterize the overall mood dynamics. On the contrary, the emotion weights directly influence the mood dynamics over the long term. Thus our idea is to find appropriate emotion weights for each personality input by optimize the above objective.

Another argument may inquire the effects of using fixed weights for different personalities. Firstly, the ALMA model by default has fixed identical weights for all emotions (all weights are 1s in ALMA). As mentioned above, the numerical simulation in section 4.4.3 shows that the ALMA model fails to approach arbitrary personalities. Secondly, the later section will prove that the expected mood will approach to the main eigenvector of a matrix with respect to the emotion weights. That shows that fixed emotion weights only work for one specific personality instead of arbitrary personalities.

4.4.2 Algorithms

In this section, we first deduct mathematical equations that describe the evolution process of the expected mood. Then we show that the expected mood will approach the main eigenvector of a matrix w.r.t the emotion weights after enough time steps. Finally we propose and solve the convex optimization problem that determine the appropriate weights for the given personalities.

Fig. 4.2 illustrates the mutual interactions among the emotions, mood and personality. The main difference of the formulation between the PCMD model and the ALMA model [45] lies in the emotion-mood interactions in the equation (4.7) that guarantees the convergence of the overall mood directions. We first revise model (4.3) into our PCMD model:

$$\mathbf{V}_M = \sum_i \alpha \mu_i \mathbf{E}_i \quad (4.7)$$

where μ_i is the (emotion) weight corresponding to the emotion with index i . All these emo-

Figure 4.2: The schema of affect dynamics

tion weights $\{\mu_i\}_{i=1}^S$ are personality dependent, controlling how each emotion contributes to the mood dynamics according to the personality.

Note that when $\mu_i = 1, i = 1, 2, \dots, S$, the model of (4.7) becomes model (4.3). However, model (4.3) cannot assure that the tendency of the mood matches the personality. Therefore we next need to carefully choose the emotion weights to achieve our goal.

Consider the mood dynamic process in an arbitrary time span, say, $[0, T]$. Denote by $N(T)$ the total number of events happened in this time span. The arrival moments of the events are denoted by $0 \leq t_0 < t_1 < t_2 < \dots < t_{N(T)} \leq T$. We suppose $N(T)$ follows a stochastic Poisson process with parameter λ , where λ represents the frequency of the event arrival. The Poisson process is widely used to model random temporal or spatial events, for example, the arrival moments of customers at a service center.

For simplicity, we suppose one event generates one emotion at one moment. The mood decay effect is not considered because the mood decays very slow and has not that much chance to decay in continuous interactions. For the event at time t_j , denote the generated emotion by e_j . The emotion direction $\mathbf{d}_j$, which the emotion weight μ_j corresponds to, is a random variable that satisfies a discrete distribution with probabilities $\{p_i\}_{i=1}^S$ over all possible emotion directions in $\mathbf{D}$. The intensity from the appraisal of the OCC model I_{j0} is a random variable that satisfies a uniform distribution $U[0, 1]$.

In the rest of the section, we start to deduct equations step by step until we finally get

the objective function of the convex optimization. We make some simplifications to make the following deductions easier. However, considering that the numerical simulations in the section 4.4.3 are conducted without any of these simplifications, the success of our model demonstrate that these simplifications are reasonable and have no significant influence on the final results.

Let consider the mood update process in the time block $[t_k, t_{k+1}]$. The mood starts from the position $M(t_k)$ and ends at the position $M(t_{k+1})$. The mood position is changed by the emotion that came in at the time t_k and the previous emotions that happened before the time t_k . Because the emotions usually decay very fast, namely δ_e is relatively large. To ease our deduction, we ignore the influence of the previous emotions that happened before the time t_k for the mood dynamics in the time block $[t_k, t_{k+1}]$. In another words, we only consider the latest emotion that came in at the time t_k . Thus the velocity of the mood in this time block is a function of the time $\tau \in [t_k, t_{k+1}]$, which is calculated as below.

$$\mathbf{V}_M(\tau) = \alpha\mu_k I_k(\tau) \mathbf{d}_k. \quad (4.8)$$

$$I_k(\tau) = [I_{k0} + \gamma_M \langle \mathbf{M}(t_k), \mathbf{d}_k \rangle + \gamma_P \langle \mathbf{P}, \mathbf{d}_k \rangle] e^{-\delta_e \tau} \quad (4.9)$$

The equation 4.8 and 4.9 come from the equation 4.7 and 4.1 respectively. Because the biasing parameters γ_M and γ_P are usually small (Both of them are 0.1 in this work, otherwise the original appraisal will be distorted too much), the biased emotion intensity in equation (4.1) falls into the boundary of $[0, 1]$ in most cases, thus we ignore the truncation of the emotion intensity $I_k(\tau)$ in later deduction.

The emotion at time t_k changes the mood position as well as decaying by itself. Here we calculate the mood dynamics in a continuous way.

$$\begin{aligned} \Delta \mathbf{M} &= \int_0^{t_{k+1}-t_k} \mathbf{V}_M(\tau) d\tau = \int_0^{t_{k+1}-t_k} \alpha\mu_k I_k(\tau) \mathbf{d}_k d\tau \\ &= \left[\int_0^{t_{k+1}-t_k} e^{-\delta_e \tau} d\tau \right] \alpha\mu_k [I_{k0} + \gamma_M \langle \mathbf{M}(t_k), \mathbf{d}_k \rangle + \gamma_P \langle \mathbf{P}, \mathbf{d}_k \rangle] \mathbf{d}_k \\ &= \left[\frac{1}{\delta_e} \left(1 - e^{-\delta_e (t_{k+1}-t_k)} \right) \right] \alpha\mu_k [I_{k0} + \gamma_M \langle \mathbf{M}(t_k), \mathbf{d}_k \rangle + \gamma_P \langle \mathbf{P}, \mathbf{d}_j \rangle] \mathbf{d}_k \end{aligned} \quad (4.10)$$

By the equation 4.4, the resulting mood position $\mathbf{M}(t_{k+1})$ is represented as:

$$\mathbf{M}(t_{k+1}) = [\mathbf{M}(t_k) + \Delta\mathbf{M}]_{-1}^1 \quad (4.11)$$

Denote

$$\mathbf{H}(t_{k+1}) = \mathbf{M}(t_k) + \Delta\mathbf{M} \quad (4.12)$$

$\mathbf{H}(t_{k+1})$ denotes the updated $\mathbf{M}(t_k)$ before normalization. If the mood position exceeds the boundary of the PAD space, we normalize the mood by dividing its infinity norm which is the absolute maximum coordinate. We replace the original mood truncation function $[\cdot]_{-1}^1$ in equation (4.4) because it changes the mood direction by only truncating the largest component of mood. The normalization is below:

$$\mathbf{M}(t_{k+1}) = \begin{cases} \mathbf{H}(t_{k+1}) / \|\mathbf{H}(t_{k+1})\|_{\infty}, & \|\mathbf{H}(t_{k+1})\|_{\infty} > 1 \\ \mathbf{H}(t_{k+1}), & \|\mathbf{H}(t_{k+1})\|_{\infty} \leq 1 \end{cases} \quad (4.13)$$

With the preparations above, we can start to compute the expectation of the mood position $E(\mathbf{M}(t_{k+1}))$. Based on the property of the Poisson process, the time interval between two adjacent events follows the exponential distribution with parameter λ , so the expectation is $E[t_{k+1} - t_k] = 1/\lambda$. The expectation of emotion direction is $E[\mu_k \mathbf{d}_k] = \sum_{i=0}^S \mu_i p_i \mathbf{D}_i$, and the expectation of the appraised emotion intensity is $E[I_{k0}] = 0.5$. Based on the equation 4.10, we first compute $E[\Delta\mathbf{M}]$ below.

$$\begin{aligned}
E[\Delta \mathbf{M}] &= E \left[\frac{1}{\delta_e} (1 - e^{-\delta_e(t_{k+1}-t_k)}) \alpha \mu_k [I_{k0} + \gamma_M \langle \mathbf{M}(t_k), \mathbf{d}_k \rangle + \gamma_P \langle \mathbf{P}, \mathbf{d}_j \rangle] \mathbf{d}_k \right] \\
&= E \left[\frac{\alpha}{\delta_e} (1 - e^{-\delta_e(t_{k+1}-t_k)}) \right] E [\mu_k \mathbf{d}_k [I_{k0} + \gamma_M \langle \mathbf{M}(t_k), \mathbf{d}_k \rangle + \gamma_P \langle \mathbf{P}, \mathbf{d}_j \rangle]] \\
&= \left[\frac{\alpha}{\delta_e} (1 - e^{-\frac{\delta_e}{\lambda}}) \right] \left[\sum_{i=0}^S \mu_i p_i \mathbf{D}_i [0.5 + \gamma_M \langle E[\mathbf{M}(t_k)], \mathbf{D}_i \rangle + \gamma_P \langle \mathbf{P}, \mathbf{D}_i \rangle] \right] \\
&= a \sum_{i=0}^S \mu_i p_i \mathbf{D}_i [0.5 + \gamma_M \langle E[\mathbf{M}(t_k)], \mathbf{D}_i \rangle + \gamma_P \langle \mathbf{P}, \mathbf{D}_i \rangle]
\end{aligned} \tag{4.14}$$

$$a = \frac{\alpha}{\delta_e} (1 - e^{-\frac{\delta_e}{\lambda}}) \tag{4.15}$$

In addition, denote $\boldsymbol{\mu} = [\mu_1, \mu_2, \dots, \mu_S]^T$, $\mathbf{p} = [p_1, p_2, \dots, p_S]^T$ and $\mathbf{1}_S = [1, 1, \dots, 1]^T$ as column vectors. $\Lambda(\boldsymbol{\mu p})$ is a diagonal matrix with the main diagonal contains the entries of the vector $\boldsymbol{\mu p}$. We represent the equation 4.14 in the form of matrix.

$$\begin{aligned}
E[\Delta \mathbf{M}] &= a \mathbf{D} \Lambda(\boldsymbol{\mu p}) [0.5 \cdot \mathbf{1}_S + \gamma_M \mathbf{D}^T E[\mathbf{M}(t_k)] + \gamma_P \mathbf{D}^T \mathbf{P}] \\
&= \gamma_M a (\mathbf{D} \Lambda(\boldsymbol{\mu p})) \mathbf{D}^T E[\mathbf{M}(t_k)] + [0.5 a \mathbf{D} \Lambda(\boldsymbol{\mu p}) \mathbf{1}_S + \gamma_P a (\mathbf{D} \Lambda(\boldsymbol{\mu p})) \mathbf{D}^T] \mathbf{P} \\
&= \gamma_M a \mathbf{U} E[\mathbf{M}(t_k)] + \mathbf{y}
\end{aligned} \tag{4.16}$$

$$\begin{aligned}
\mathbf{U} &= \mathbf{D} \Lambda(\boldsymbol{\mu p}) \mathbf{D}^T, \quad \mathbf{y} = 0.5 a \mathbf{D} \Lambda(\boldsymbol{\mu p}) \mathbf{1}_S + \gamma_P a \mathbf{U} \mathbf{P} \\
\Lambda(\boldsymbol{\mu p})_{j,j} &= \mu_j p_j, j = 1, 2, \dots, S
\end{aligned} \tag{4.17}$$

$\mathbf{y}$ is a vector with respect to the time t_k and $\mathbf{U}$ is a symmetric matrix. By integrating with the equation 4.12, we have the dynamics of $E(\mathbf{H}(t_{k+1}))$ below.

$$\begin{aligned}
E[\mathbf{H}(t_{k+1})] &= E[\mathbf{M}(t_k) + \Delta\mathbf{M}] = E[\mathbf{M}(t_k)] + E[\Delta\mathbf{M}] \\
&= E[\mathbf{M}(t_k)] + \gamma_M a \mathbf{U} E[\mathbf{M}(t_k)] + \mathbf{y}
\end{aligned} \tag{4.18}$$

From the equation 4.18, there are three terms that constitute to the $E(\mathbf{H}(t_{k+1}))$. The first term $E[\mathbf{M}(t_k)]$ is the mood expectation at the beginning of the time block, while the last two terms contribute to the dynamics. The second term performs an effect of scaling and rotation to $E[\mathbf{M}(t_k)]$, while the third term adds a constant vector. By combining with the equation 4.13, we analyze the convergence of the mood expectation $E[\mathbf{M}(t_k)]$ below. Firstly, let discuss two simplified cases. On one hand, if we remove the second term $\gamma_M a \mathbf{U} E[\mathbf{M}(t_k)]$ while keep the constant vector $\mathbf{y}$, it is clear that $E[\mathbf{M}(t_k)]$ will converge to the direction of $\mathbf{y}$ after enough time steps. On the other hand, if we remove the third term while keep the second term, the $E[\mathbf{M}(t_k)]$ is scaled and rotated by the symmetric matrix $\mathbf{I} + \gamma_M a \mathbf{U}$ ($\mathbf{I}$ is an identical matrix). Based on the power method proved by [92], the $E[\mathbf{M}(t_k)]$ will converge to the main eigenvector of the matrix $\mathbf{I} + \gamma_M a \mathbf{U}$.

However the combination of the last two terms makes the proof of convergence much more complicated. Here we just indicate the final vector that the $E[\mathbf{M}(t_k)]$ will converge to after enough time step. Informed by the power method, let try to reformulate the equation 4.18 in the form that $E[\mathbf{M}(t_k)]$ is multiplied by a matrix in every time step. Firstly we transform the constant vector $\mathbf{y}$ to be the multiplication of a matrix $\mathbf{R}_k$ and the vector $E[\mathbf{M}(t_k)]$.

$$\mathbf{y} = \mathbf{R}_k E[\mathbf{M}(t_k)] \tag{4.19}$$

$\mathbf{R}_k$ could be the matrix of a Householder transformation, which is a linear transformation that describes a reflection about a plane or hyperplane. The transformation is given below.

$$\mathbf{R}_k = \frac{\|\mathbf{y}\|}{\|E\mathbf{M}(t_k)\|} (\mathbf{I} - \mathbf{w}\mathbf{w}^T), \quad \mathbf{w} = [\mathbf{y}]^\circ - [E\mathbf{M}(t_k)]^\circ \tag{4.20}$$

Then the equation 4.18 can be reformed below.

$$E[\mathbf{H}(t_{k+1})] = (\mathbf{I} + \gamma_M a \mathbf{U} + \mathbf{R}_k) E[\mathbf{M}(t_k)] \quad (4.21)$$

If the $E[\mathbf{M}(t_k)]$ converges to the vector $\mathbf{M}^*$, then the matrix $\mathbf{R}_k$ converges to a matrix $\mathbf{R}$ by replacing $E[\mathbf{M}(t_k)]$ with $\mathbf{M}^*$ in the equation 4.20. In addition, $(\mathbf{I} + \gamma_M a \mathbf{U} + \mathbf{R}) \mathbf{M}^*$ must have the same direction with $\mathbf{M}^*$. In another words, by the power method, $\mathbf{M}^*$ must be the main eigenvector of the matrix $(\mathbf{I} + \gamma_M a \mathbf{U} + \mathbf{R})$.

Afterwards, we need to make the converged mood expectation $\mathbf{M}^*$ equals to the personality $\mathbf{P}$. Instead of finding the most appropriate weights $\{\mu_i\}_{i=1}^S$ to make $\mathbf{P}$ become the main eigenvector of the matrix $(\mathbf{I} + \gamma_M a \mathbf{U} + \mathbf{R})$ considering that $\mathbf{R}$ is hard to compute due to its dependence on $\mathbf{M}^*$, we adopt a simpler approximation method by making $\mathbf{M}^*$ approximately equals to $\mathbf{y}$ as well as making $\mathbf{y}$ equals to $\mathbf{P}$. Let consider the ratio ρ of the norms of $\gamma_M a \mathbf{U}$ and $\mathbf{R}_k$ as below.

$$\rho = \frac{\|\gamma_M a \mathbf{U}\|}{\|\mathbf{R}_k\|} = \frac{\gamma_M a \|\mathbf{U}\| \|\mathbf{M}_k\|}{\|\mathbf{y}\|} \leq \frac{\gamma_M a \|\mathbf{U}\|}{\|\mathbf{y}\|} \quad (4.22)$$

If $\mathbf{R}_k$ has significant larger norm than $\gamma_M a \mathbf{U}$ (i.e., $\rho < 1$), then $\mathbf{R}_k$ contributes to the main dynamics of the $E[\mathbf{M}(t_k)]$. In another words, the second term $\gamma_M a \mathbf{U}$ is neglectable after enough time steps. As analyzed above, the final vector $\mathbf{M}^*$ is approximately equals to $\mathbf{y}$. Meanwhile, we require $\mathbf{y}$ equals to $\mathbf{P}$. The expected mood direction will approach the personality direction in long enough time span with an arbitrary initial mood. Although we don't provide rigorous proof in mathematics, the validity of this method can demonstrated by the numerical simulations in section 4.4.3.

Thus the objective function is constructed as a convex optimization problem:

$$\begin{aligned} \arg \min_{\{\mu_i\}_{i=1}^S} \quad & \phi_1 \|\mathbf{y} - \mathbf{P}\|^2 + \phi_2 \|\mathbf{U}\|_F^2 \\ \text{s.t.} \quad & \mu_i > 0, i = 1, 2, \dots, S, \end{aligned} \quad (4.23)$$

where $\|\mathbf{y} - \mathbf{P}\|$ measures the difference between y and $\mathbf{P}$, $\|\mathbf{U}\|_F$ is the Frobenius norm of matrix $\mathbf{U}$ that controls the norm of the second term $\gamma_M a \mathbf{U}$, and ϕ_1 and ϕ_2 are tradeoff factors (in this work, we set $\phi_1 = 1, \phi_2 = 10^{-2}$). The above optimization problem is solved by the solver “fmincon” provided by Matlab. Because the problem (4.23) is convex, the objective has global minimal value. The considered types of emotions can be selected by the users in advance by setting the emotion direction matrix $\mathbf{D}$. Given certain personality, the optimization problem (4.23) only needs to be solved once at the preprocessing process to obtain the required emotion weights.

4.4.3 Numerical Simulation

This section presents numerical simulations for the mood dynamics given randomly generated events and personalities. The PCMD model is compared with the baseline model (ALMA), which is the degenerated case where all emotion weights are equal to 1. We implement the mood dynamics without considering the simplifications proposed in the section 4.4.2. The main difference between the two models is the setting of the emotion weights. We want to figure out the following problems:

- (1) As shown in Fig. 4.2, although the baseline model doesn’t differentiate the weights for emotions directly, the personality also influences the intensities of emotions, which further updates the moods. Thus, will the indirect influence of the personality on the mood guarantee the overall mood tendency or not?
- (2) Can the convergence of the equation (4.23), which is not strictly proved in the section 4.4.2, be demonstrated as valid and stable over any personalities, initial moods and event sequences or not?
- (3) Do the simplifications proposed in the section 4.4.2 make much difference or not?
- (4) How fast is the convergence of the PCMD model?

After sufficient interactions (say 1000 events), we measure the convergence of the mood tendency to the personality by the **overall direction closeness (ODC)** between them, which is defined as the average of the cosine values of the angles between the two vectors in the process of interactions. The range of the cosine value falls in $[-1, 1]$. The larger the value is, the closer the mood and personality are.

Design and Procedure Two experiments are conducted. Besides the overall direction convergence of the models, experiment 1 focuses on the stableness over the randomly generated events given certain personality and initial mood. Experiment 2 focuses on the stableness over the randomly generated personalities and initial moods given certain sequences of events.

In these experiments, the event arrival time, the corresponding emotion types and initial intensities are randomly generated as discussed in Section 4.4. The parameters are set as follows: $\alpha = 2$, $\gamma_M = 0.1$, $\gamma_P = 0.1$, $\delta_e = 1$, and $\lambda = 1/3$.

Exp. 1 tests the models 200 times in parallel. Each time we fix the personality and the initial mood, which are randomly generated:

$$\mathbf{P} = (0.2343, -0.7720, -0.0737)^T,$$

$$\mathbf{M}_0 = (0.5365, 0.4661, -0.3140)^T,$$

while randomly generate 1000 events over the time span of 3000 seconds. We compute the variation of moods and examine the overall mood direction against the input personality direction.

Exp. 2 also tests the models 200 times in parallel. We fix a sequence of randomly generated 1000 events. Each time the personality and initial mood are randomly generated as the input.

Results and Analysis The mean and the standard errors of the ODC values over the 200 tests in Exp. 1 and Exp 2 are presented in Table 4.5.

Table 4.5: Statistics of the ODC values

	Baseline Model	PCMD Model
Exp. 1	M=-0.657, SD=0.073	M=0.994, SD=0.0006
Exp. 2	M=0.095, SD=0.557	M=0.993, SD=0.006

(a) PCMD Model

(b) Baseline Model

Figure 4.3: The results of the cosine values between the current mood and the personality in the first test

According to the first column of Table 4.5, the baseline model fails to guarantee the convergence of the overall mood tendency to personality, since the means of the ODC values of the baseline model in both experiments are far from 1. The non-convergence results from the weak and indirect personality-mood relationship in the baseline model. In addition, by considering the standard errors of the model, it seems that the baseline model is relatively stable over the random event sequence but not over the various personalities and initial moods. Referring to the second column of the Table 4.5, the PCMD model is both convergent and stable with the observation that the means of the model is close to 1 and the standard errors is almost zero in both experiments. Moreover, by T-test (for mean) and F-test (for standard error), we found that the PCMD model is significantly more stable than the baseline model over random personalities and events. The reason for the difference in

results is that the PCMD model assigns different weights to emotions for updating the mood given certain personalities, while the baseline model treats all emotions in equal. Because the emotion weights obtained by the PCMD model works well on the mood dynamics process without the simplifications proposed in the section 4.4.2, we can conclude that the simplifications, which are proposed for simpler mathematical deduction, makes not much difference to the final results.

To illustrate the simulation process of interaction, we display the process of mood dynamics in detail for the first test in Exp. 1. Fig. 4.3 displays the dynamics of the cosine value of the current mood and the personality with time. The first graph in Fig. 4.3 illustrates that the cosine value of the PCMD model increases and first arrives at 1 after nearly 450 seconds (after 150 events), then the cosine value fluctuates slightly but keeps in high level. On the other hand, the cosine value of the baseline model shown in the second graph doesn't converge and fluctuate heavily. The convergence of the PCMD model seems not very fast, as a limitation, which is partially due to the high randomness and variety of events.

4.5 Experiments for Example Interactions

In this section, we presents four examples to evaluate the psychological plausibility of the PCMD model in comparison with the ALMA model [45]*. In each example, the dynamics of the pleasure, arousal and dominance of the mood are visualized by the curves in three sub-figures. In each sub-figure, three curves in blue, red and green colors represent the mood dynamics given three types of personalities respectively: Optimistic, Neutral and Pessimistic (see Table 4.6). The settings of the personalities are shown in Table 4.7. While the mood in the ALMA model starts at the personality position, the mood in the PCMD model purposely starts at the origin of the PAD space to show that the initial position of the mood is not very important in the new model for longer-term interaction.

*We use the source codes that are provided by the author and accessible from the website: <http://www.dfki.de/gebhard/>

Table 4.6: Three types of personality

Personalities	O	C	E	A	N
Optimistic	0.4	-0.4	0.1	0.7	0.1
Neutral	0	0	0	0	0
Pessimistic	-0.2	0.4	-0.1	-0.5	-0.7

Table 4.7: The PAD settings of the three personalities

Personalities	P	A	D
Optimistic	0.45	0.21	-0.13
Neutral	0	0	0
Pessimistic	-0.45	0.22	0.12

The parameters used in the PCMD-algorithm are set as follows: $\lambda = 1/3, \alpha = 2, \delta_e = 1, \gamma_M = \gamma_P = 0$. The parameters used in the ALMA model are set as follows: Mooddecaytime = “1000000”, Mooddecayperiod = “100”, Emotiondecaytime = “50000”, Emotiondecayperiod = “100” Emotiondecayfunction = “exponential”, RealtimeOutputperiod = “100”.

Example 1 A negative event generating an emotion “anger” with intensity 0.5 happens at the beginning. The dynamics of the mood position is shown in Fig. 4.4. In the PCMD model, the pessimistic companion has the biggest changes to the negative emotion “anger”, while the optimistic companion has the smallest changes. In the ALMA model, the change rates of the mood positions have no difference for the three personalities.

Example 2 A positive event happens at the beginning, which generates an emotion “gratitude” with intensity 0.5. Fig. 4.5 shows the dynamics of the mood position. In the PCMD model, the optimistic companion has the biggest changes to the positive emotion “gratitude”, while the pessimistic companion has the smallest changes. In the ALMA model, the mood positions are different while the change rates of the mood positions are the same for the three personalities.

Figure 4.4: Mood dynamics for an anger emotion

Figure 4.5: Mood dynamics for a gratitude emotion

Example 3 A negative event generating an emotion “anger” with intensity 0.5 happens at the beginning, followed by a positive event generating an emotion “gratitude” with intensity 0.5. The dynamics of the mood position is shown in Fig. 4.6. In the PCMD model, the optimistic companion has the biggest changes to the positive emotion “gratitude” and the final mood position shares the same octant with the emotion “gratitude” in the PAD space, while the pessimistic companion has the biggest changes to the negative emotion “anger” and the final mood position shares the same octant with the emotion “anger”. In the ALMA model, the tendencies of the mood positions are similar for the three personalities. The mood of

the pessimistic companion is always negative even though the positive event happens at the beginning.

Figure 4.6: Mood dynamics for an anger emotion followed by a gratitude emotion

Example 4 A positive event generating an emotion “gratitude” with intensity 0.5 happens at the beginning, followed by a negative event that generates an emotion “anger” with intensity 0.5. The dynamics of the mood position is shown in Fig. 4.7. In the PCMD model, the optimistic companion has the biggest changes to the positive emotion “gratitude” and the final mood position shares the same octant with the emotion “gratitude” in the PAD space. The pessimistic companion has the biggest changes to the negative emotion “anger” and the final mood position shares the same octant with the emotion “anger”. In the ALMA model, the tendency of the mood positions is similar for different personalities. The mood of the optimistic companion is always positive though the negative event happens at the beginning.

Discussion In the above four examples, the mood dynamics of the optimistic, neutral and pessimistic personalities are represented in blue, red and green colors. The inflection point appears when a new emotion is processed. In the PCMD-algorithm, the initial mood starts at the neutral point because nothing has happened before. The optimistic companion is more

Figure 4.7: Mood dynamics for a gratitude emotion followed by an anger emotion

sensitive to the positive information while the pessimistic companion is more sensitive to the negative information. The difference of the three personalities can be indicated by the change rate of the mood to different emotional stimuli during the interactions. In the ALMA model, the tendencies of the mood dynamics for the three personalities keep the same, although the initial mood positions are different. Hence the difference of the personalities in the ALMA model is reflected by the mood position only at the very beginning, rather than by the tendency of the mood dynamics during the interactions. However, in longer-term interactions, because the mood position can be arbitrary at any time during the interaction, the tendency of the mood dynamics is more important to show particular personalities.

4.6 Case Study

The PCMD model is integrated with our virtual human named Sophie to enhance the predictability of the personalities from her affective behavior patterns. The system structure is shown in Fig. 4.8. The system integrates many functionalities including natural language processing, face recognition, affective engine, facial expression, gesture generation and text-to-speech modules.

Figure 4.8: The system structure of the virtual human

4.6.1 Settings

In the application scenario, Sophie works as a receptionist for the users to answer their questions about her lab. There are two kinds of users, the polite user or the rude user, shown in Fig. 4.9. After each iteration of question and answer, the polite user makes positive comments (like “Your explanation is very clear”) on Sophie which result in “gratitude” emotions, while the rude user tends to make negative comments (like “Sorry, your accent is too strange”) which lead to “anger” emotions. To make comparison, we design two opposite personalities, optimistic or pessimistic, for Sophie, of which the settings are the same with the Table 4.6 and Table 4.7. The rules of the emotional behavior selection based on the current emotions and moods are predefined in scripts. The available emotional behaviors and the designed rules of behavior selections are the same for the two groups so that the different behavior patterns are mainly driven by the pattern of moods.

With the PCMD model, Sophie tends to react more and more aggressively to the same comments when her mood is getting worse, while showing great tolerance to the negative comments when her mood is pleasant. For example, when the users complain that they fail

to understand her (that generates “anger” emotions), Sophie will apologize for the misunderstandings (“Sorry, I can speak slower”) if the her mood is pretty good; Otherwise, she will blame the users for their distraction (“I hope you can keep your attention”). When the virtual human is extremely angry, she will refuse to answer any questions (“Nothing to say”), unless the users try to comfort her (“Do you feel better”) or wait until she calms down by herself. The mood also influences the non-verbal behaviors of Sophie. For example, if Sophie is in a bad mood and the users make a negative comment, she will show angry facial expressions and intensive gestures. However, if the users make a negative comment when Sophie is in a pretty good mood, she will slightly smile to show her forgiveness for the undesirable event instead of showing an angry face immediately. The facial expressions, gestures and the speech tones are chosen based on the types and intensities of the current mood and the input emotions. Sophie chooses more powerful gestures if her mood is intensive.

4.6.2 Results and Discussion

To make an evaluation, we consider the interaction scenarios of the optimistic or pessimistic Sophie with the polite and the rude users in different order. The interaction results inform us the following observations. Firstly, Sophie reacts flexibly to the various situations instead of keeping rigid behavior strategies. In the scenarios, when the polite user keeps making positive comments on Sophie, she becomes exuberant even though she has pessimistic personality. On the other hand, continuous negative comments from the rude user also irritate the optimistic Sophie. Secondly, although Sophie has flexible behavior strategies, her overall behavior pattern keeps consistent. Comparing the performance of the same Sophie with the polite and the rude users with different orders, the behaviors of Sophie showed certain tendency. For example, when the optimistic Sophie first interacted with the rude user, she showed slight angry behaviors at first but quickly recovered by the later interactions with the polite user. On the other hand, when she first interacted with the polite user, she quickly

(a) Polite User

(b) Rude User

Figure 4.9: Sophie interacts with polite user and rude user.

came into a highly exuberant state which was not much influenced by the rude user later. Overall, the optimistic Sophie showed similar behavior tendency in both cases. Thirdly, the personality of Sophie is reflected from her behaviors in the interactions. One evidence is that by comparing the performance of the two Sophies, it is easier and faster for Sophie with optimistic personality to move from the negative mood to the positive one, than to move from the positive mood to the negative one, and vice versa. Moreover, generally speaking, the optimistic Sophie shows much more exuberant expressions, friendly gestures and positive feedbacks than the depressed expressions, rude gestures and negative feedbacks. To obtain a subjective evaluation, we videotaped the interaction process and showed them to the recruited participants. Many participants report that they can easily recognize the exact personality of Sophie.

4.7 User Study

The user study is conducted in the context of the application scenario in Section 4.6. In this study, we make comparison of the ALMA model and the proposed PCMD model on the same platform with our social robot Nadine to investigate several aspects of companion-user interactions. Firstly, we are interested in whether the affective behavior patterns can be identified. Secondly, we investigate the easiness and correctness of personality recognition.

Figure 4.10: Interaction with Nadine with the PCMD model or the ALMA model

4.7.1 Design and Procedure

The within-subject user study compares the PCMD model with the ALMA model [45][†], which is selected because the two models have similar affective framework except for the overall mood convergence. To make fair comparisons, the two models share the same social robot platform, user inputs, available behaviors and equivalent parameter settings. Although the ALMA model can achieve good effects of personality simulation at the very beginning by initializing the mood by the personality, we are interested in whether the two models can produce consistent personalized behavior patterns to convey the intended personalities during the whole interactions. Hence we simulate a medium period of the interactions that the mood starts from neutral at the very beginning.

Four versions of Nadine have been developed: a) Optimistic Nadine with PCMD model (PCMD + P); b) Optimistic Nadine with ALMA model (ALMA + P); c) Pessimistic Nadine with PCMD model (PCMD + N); d) Pessimistic Nadine with ALMA model (ALMA + N). The settings of the optimistic and pessimistic personalities are the same with the Table 4.6 and Table 4.7. We created four videos to the interactions with the four versions of Nadine respectively. In each video, a user alternatively makes positive comments and negative comments on Nadine. The positive comments can generate the “gratitude” emotions

[†]We use the source codes that are provided by the author and accessible from the website: <http://www.dfki.de/gebhard/>

Figure 4.11: Comparison performance in emotion sensitivity

while the negative comments can generate the “anger” emotions. To compare the effects of personalities, we set the emotions with the same intensities. The order and the number of the positive or negative comments in each video are the same. The interaction scenes of Nadine of different versions are shown in Fig. 4.10.

Upon arrival, we give the participants a brief introduction of the procedure of the experiment and we collect some of their personal information. They are required to fill in a form about their previous experience with the social robots and the affective computing. Each participant is asked to sit beside a table and wear a headphone set to watch the four videos. The order of the four videos are adapted by the Latin square rotation to reduce the bias of learning or the fatigue of the participants. After each video, participants are required to complete a questionnaire, shown in the Appendix A, with 7-Likert scales to indicate how they perceived the behavior patterns of Nadine and her personalities. The questions for the four versions of Nadine are the same.

Figure 4.12: Comparison performance in recognition easiness

4.7.2 Results

A total of 24 participants, 8 female and 16 male with an average age of 28 ranging from 23 to 34, were recruited for this study. 37.5% of the participants were familiar with social robot, and the score for familiarity with affective computing was 12.5%. Figure 4.11, Figure 4.12 and Figure 4.13 display the results of user ratings for the four versions of Nadine with respect to the emotion sensitivity, the recognition easiness and the personality polarity respectively. From the analysis of the data displayed in Table 4.8 using the two-tail T-test, we obtain the following results. The participants had the observation that the optimistic Nadine with the PCMD model behaved more sensitively to the positive comments than the one with the ALMA model ($t = 9.333, t(23) = 2.807, p = 0.01$), while the pessimistic Nadine with the PCMD model behaved more sensitively to the negative comments than the one with the ALMA model ($t = 7.897, t(23) = 2.807, p = 0.01$). For the easiness of the personality recognition, we find that the personality of the optimistic Nadine with

Figure 4.13: Comparison performance in personality polarity

Table 4.8: Experimental results of the user study

Measures	Optimistic Nadine		Pessimistic Nadine	
	PCMD	ALMA	PCMD	ALMA
Emotion Sensitivity	M=4.96, SD=1.40	M=0.50, SD=1.84	M=4.21, SD=2.43	M=0.08, SD=0.65
Recognition Easiness	M=6.50, SD=0.66	M=3.92, SD=1.41	M=5.88, SD=1.54	M=3.75, SD=1.75
Personality Polarity	M=6.58, SD=0.50	M=4.50, SD=0.72	M=1.96, SD=1.04	M=4.13, SD=0.45

the PCMD model is evaluated to be more easily recognized than the one with the ALMA model ($t = 8.598, t(23) = 2.807, p = 0.01$), and the same for the pessimistic Nadines ($t = 5.245, t(23) = 2.807, p = 0.01$). In addition, participants also have better identified the personality of the optimistic Nadine with the PCMD model than the one with the ALMA model ($t = 13.160, t(23) = 2.807, p = 0.01$), and so does the pessimistic Nadine with the PCMD model ($t = 9.737, t(23) = 2.807, p = 0.01$).

4.7.3 Discussion

The result that the high overall emotion sensitivity has been observed in Nadine with our PCMD model is because the PCMD model assigns more weights for the emotions that are consistent with the personality in the mood dynamics. For example, the optimistic Nadine with the PCMD model is sensitive to the positive comments because the “gratitude” emotion has larger weights than the “anger” emotion. Hence Nadine tends to keep her good moods as well as to show great tolerance to the bad events, while the pessimistic Nadine tends to understand events negatively. However, the ALMA model puts no direct bias in mood dynamics given particular personalities. Therefore, Nadine with the ALMA model displays no clear emotional sensitivity as given the positive and negative inputs alternatively.

In addition, the results indicate that our PCMD model enables the personalities of Nadine to be more distinctive to be recognized. This informs us that the clear emotion sensitivity observed in the behavior patterns help users to differentiate personalities. Therefore, once reasonable rules of the mood-driven behavior selection are built, Nadine has the potential to simulate various personalities by the appropriate patterns of the mood dynamics. On the contrary, the unclear emotional sensitivity of the ALMA model results in less observed distinctiveness of personalities in interactions.

Finally, the polarity of the particular personalities of Nadine with the PCMD model can be correctly identified, while the personalities of Nadine with the ALMA model are observed to be more neutral. This indicates that the PCMD model has achieved correct and observable personality effects, of which the polarity aligns with that of the overall emotion sensitivity.

Chapter 5

MCAEM: Mixed-Correlation-Analysis based Episodic Memory

In this chapter, we present a mixed-correlation-analysis based episodic memory (MCAEM) to improve retrieval accuracy. We introduce the concept of mixed correlation analysis, which describes the memory elements in three aspects: the relations between elements, the weights of attributes, and the order of events. Based on the mixed correlation analysis, a new similarity measure is proposed, which has substantially improved the performance of the model. To make evaluation, we have applied the MCAEM model to both intelligent agents in computer games and social companion interactions with free dialog.

5.1 Introduction

Episodic memory models focus on encoding, storing and retrieving past experience and how to use retrieved results for further decision making. Retrieval is the core functionality of episodic memory. It provides fundamental support for higher-level cognitive capabilities in decision making. Memory models retrieve events or episodes that are most relevant to given cues. Thus, to improve retrieval accuracy, prior studies take many efforts in modeling the event-cue and episode-cue similarity. For example, the number of shared elements

Figure 5.1: The four aspects of the similarity measure in episodic memory

between events and cues or episodes and cues is one of the well-discussed factors for similarity measures in prior work. The more shared elements between events or episodes and cues, the more relevant they are. In this chapter, we propose a mixed correlation analysis which models correlations of memory elements (attributes, events and episodes) in many aspects. The mixed correlation analysis can enable more efficient memory organization and provide better similarity measure to improve retrieval accuracy.

As shown in Figure 5.1, the mixed correlation analysis of memory elements includes two types: content correlation and order correlation. The content correlation further considers the relations between elements and the weights (importance) of attributes. And the order correlation considers the order of events in episodes. We discuss the three types of correlations below.

Elements of episodic memory have three levels: attributes, events and episodes. The memory elements have relations with each other of different intensities and types. Taking attribute-attribute relations as example, there exists two types of relations: the *relations between classes* and the *relations within classes*. The *relations between classes* consider some attributes of different types that always coexist in same contexts. Taking the relations of food and restaurants as an example, Figure 5.2 (a) displays that the Big Mac always appears with the McDonald's, the bread with egg and cheese always appears with the Subway,

Figure 5.2: Examples for two types of attribute-attribute relations

and the coffee (the special one) always appears with the Starbucks. On the other hand, the *relations within classes* consider some attributes of same classes that can be replaced by each other with similar semantic meanings. For example, Figure 5.2 (b) displays four kinds of food in the McDonald's, the McSpicy is more similar to the Big Mac than the fries or the McCafe. Therefore, I can accept the McSpicy instead if the Big Mac is sold out today because they are similar to each other.

It is important to model the relations between memory elements in the same levels or across different levels. Some prior studies have considered relations across different levels [135], such as attribute-event and event-episode relations, however the relations in the same levels (attribute-attribute, event-event and episode-episode) are rarely modeled or utilized for memory organization or retrieval. Our work analyze both relations in same levels and relations across different levels.

Besides the relations between elements, another content correlation is the weights or importance of attributes in different episodes. Instead of treating each attribute with equal importance, attributes should be given different weights at particular contexts. Taking the words in news reports as example, Figure 5.3 shows that words have different weights in new reports of particular topics. The words in bigger sizes are more important than the

relational space that maps every memory element as a vector and computes their relations using distances. To analyze the weights of attributes, we apply the TF-IDF (term frequency-inverse document frequency) method [110] to include both the occurrence frequency and the specialness of attributes in particular episodes. For the order correlations of episodes, we propose a sliding-window algorithm which is balanced and robust to positions of events. Finally we propose a computational architecture to measure event-cue or episode-cue similarities by combining the above three correlations.

The proposed MCAEM model is integrated with both intelligent agents and social companions. To evaluate its learning ability, the MCAEM model is applied to a first-person shooting game, Unreal Tournament 2004 (UT2004), to improve agent performance by learning from past experience. To improve the user engagement in companion-user interactions, the MCAEM model is integrated with our social companions to provide memory functionalities in free-dialog based social companion interactions. The success in the two distinct applications demonstrates effectiveness and generality of our MCAEM model.

5.2 Related Work

Various computational models of episodic memory have been proposed in recent years for intelligent agents and social companions. Some early studies are designed for specific application domains [61, 65]. Recent research shows great interest in domain-independent models for general purpose [98, 135]. An episodic memory model should provide accurate and efficient retrieval. The retrieval performance relies on good analysis of correlations of memory elements which can provide better measure of their similarities with cues. In this section, we review kinds of correlations of memory elements and similarity measures in prior studies.

Some studies have started to consider the relations between elements. Brom et.al [13] consider the relations between locations and objects and Gomes et.al [51] connect emotions with locations. However, it is difficult to extend these methods to handle general attributes.

Some studies try to make connections with semantic memory which contains attribute relations. Tecuci and Porter [122] retrieve episodes by two steps: a shallow exact matching for the attributes followed by a deep matching for the graph structures of episodes. They adopt semantic similarities of attributes in the deep matching, where the semantic matcher needs to represent episodes as graphs and to annotate causal relations among concepts. This approach is time-consuming and not general enough for raw memory data. Mueller and Shiffrin [95] build connections of episodic memory and semantic memory. However, the feature co-occurrence matrix demanded for the semantic memory for each concept is too tedious to extract the relations for all attributes. Different from these work, our proposed episodic memory model is domain-independent and supports effective correlation extraction of attributes, events and episodes.

Some prior studies consider attributes with equal importance. Thus they provide similarity measures based on number of shared elements which returns episodes that share the most attributes with cues [98]. Some studies retrieve episodes based on the average [105] or the weighted average [33] of the shared attributes. The weights of attributes include their frequency of occurrence, emotional saliences or importances for goals or attentions [33, 30, 70]. However, all these similarity measures have few consideration of the specialness of attributes in episodes.

Some studies also consider the order of event sequences. Faltersack et.al [40] propose Ziggurat memory model that searches for the longest sequence matching between the most recent subsequence of memories and the remaining memory. However, in the Ziggurate model, searching for the remaining memory each time is very time-consuming in large-scale memory database. Jockel et.al [65] propose EPIROME memory model that compares order of events in episodes by Viterbi-algorithm [132]. Viterbi-algorithm is a dynamic programming algorithm designed to find the most likely sequence of hidden states. The EPIROME model use the Viterbi-algorithm to compute probabilities (regarded as similarities) that one episode can be generated by another episode. However the EPIROME model may be slow because it fails to optimize the Viterbi-algorithm for some cases of retrieval

(for example, the cases that cues are much shorter than episodes).

The EM-ART model is the most relevant to our work. The authors have published groups of work [135, 136] in recent years with the core memory framework keeping consistent. In their work, episodic memory is modeled by a Fusion-ART neural network with three hierarchies for attributes, events and episodes respectively. The EM-ART model learns the key events and episodes by online clustering and the connection weights of memory elements across different levels. The weights of elements are updated based on how frequent the element appears in particular clusters. However, the specialness of the elements is not considered in their work. Although they have considered the attribute-event and event-episode relations by aggregating the learned weights of shared elements with given cues, the attribute-attribute relations for event retrievals and the event-event relations for episode retrievals have not been considered. Finally, they measure the event-order similarity by the sequence of event activations which decay with time exponentially. Thus recent events are given much more higher activations than earlier events. Therefore, this method biases events of different positions. This is because matching with a subsequence of an episode at the beginning gives less help in retrieval than matching a single event at the end. Our work is aimed at analyzing correlations of memory elements that can provide better memory organization and more accurate similarity measure. We make comparison with the EM-ART model in experiments because we share the similar hierarchical structure and the same platform for validation.

Brief Introduction to the EM-ART Model

We first give a brief introduce to more details of the EM-ART model [135]. The EM-ART model has three layers for attributes, events and episodes respectively. Figure 5.4 displays the overall structure of the EM-ART model. The F_1 , F_2 and F_3 are attribute layer, event layer and episode layer respectively.

EM-ART model is an integration of two ART channels. Taking the first channel as an example, as shown in Figure 5.5, this channel has two layers: attribute layer F_1 and event

Figure 5.4: The overall structure of EM-ART model

layer F_2 . The F_1 layer has multiple input fields c_1 to c_n . The input for field k is a vector $\mathbf{x}^k = (x_1^k, x_2^k, \dots, x_{n_k}^k)$. The F_2 layer has multiple categories by online-clustering F_1 inputs. Let $\mathbf{y} = (y_1, y_2, \dots, y_m)$ be the activation vector of all F_2 categories. $\mathbf{w}_j^k$ denotes the weight vector from the j -th category of F_2 to the k -th input field of F_1 . The weights are adapted with online-learning.

The steps of training for each ART channel are below:

Code Activation Given input vector $\mathbf{x}$, compute the activations of all F_2 categories. The activation for category j is computed below:

$$T_j = \sum_{k=1}^n \gamma^k \frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha^k + |\mathbf{w}_j^k|},$$

where the fuzzy AND operation $\wedge$ is defined as element-wise minimization, and the norm $|\cdot|$ is defined by summation of all elements. $\alpha^k > 0$ is the choice parameter for input field k . $\gamma^k \in [0, 1]$ is the contribution parameter of input field k to the activation in F_2 .

Code Competition Select the most active F_2 category J by $T_J = \max\{T_j : \forall j \in F_2\}$.

By winner-take-all strategy, the activation vector $\mathbf{y}$ is set all zero except $y_J = 1$.

Template Matching Compute the matching score between the weight $\mathbf{w}_J$ of F_2 category

Figure 5.5: The structure of a single ART channel

J and the input vector $\mathbf{x}$. For each input field k , there is criterion that

$$m_J^k = \frac{|\mathbf{x}^k \wedge \mathbf{w}_J^k|}{|\mathbf{x}^k|} \geq \rho^k,$$

where $\rho^k \in [0, 1]$ is the vigilance parameter for the k -th F_1 input field. If all these criteria are satisfied, the F_2 category J is matched. Otherwise, remove J from consideration and do the step 2 again until a category is matched. If no category is matched, create a new one in F_2 with weight equals to the input vector $\mathbf{x}$.

Template Learning For the matched F_2 category J , the weight is updated by the input vector $\mathbf{x}$ by $\mathbf{w}_J^k = (1 - \beta^k)\mathbf{w}_J^k + \beta^k(\mathbf{x}^k \wedge \mathbf{w}_J^k)$, where $\beta^k \in [0, 1]$ is the learning rate parameter for the input field k .

The training of the second channel is similar to the first channel, except that the input from the event layer F_2 to the episode layer F_3 are different. There is only one input field of F_2 , the input vector is a sequence representing the order of the happened events. The later events have higher activations by the recency of time.

The retrieval process is to retrieve the most matched F_2 event categories and F_3 episode categories, if any, given F_1 input streams. The details refer to [135].

We get the source codes written by the authors in Java on the platform of Pogamut 2. In this experiment, we update the codes to the latest platform Pogamut 3. The attribute layer F_1 has four input fields: the current location, the health condition, the selected behavior and the reward. Table 5.1 displays the parameter setting for the two channels.

Table 5.1: The parameters of EM-ART model in this implementation

Channel	Field	α^k	β^k	γ^k	ρ^k
Event (F1)	Location	0.1	0	0.4	1
	Health	0.1	0	0.2	1
	Behavior	0.1	0	0.2	1
	Reward	0.1	0	0.2	1
Episode (F2)		0.1	0	1	1

5.3 The Proposed Episodic Memory

In this section, we propose a mixed-correlation-analysis based episodic memory model. We first introduce three levels of memory elements: attributes, events and episodes. Attributes are the bottom-level memory elements. They represent the objects, states and concepts that constitute events and episodes. Attributes have two properties: class and value. For example, to clearly understand the process of a general event, one needs information of the subject (who), the action (what), the object (whom), the location (whom) and the time (when). All of these are attribute classes. Nevertheless, to describe a specific event, we need to name the specific values for each attribute class. For instance, Tom (subject) talked (action) with Mary (object) at school (location) yesterday (time). Events are the middle-level memory elements. They are collections of attributes of multiple classes. We represent events as lists of attributes in the form of “Class = Value”. For example, the above event can be represented as [Sub=Tom, Act=Talk, Obj=Mary, Loc=School, Time=Yesterday]. Episodes are the top-level memory elements. They contain a sequence of events with time order. The order of events can differentiate episodes with similar contents.

In the following sections, we propose correlation analysis of memory elements in three aspects: the relations between elements, the weights of attributes and the order of events. Based on the correlation analysis, we propose a new similarity measure. Finally, we present our episodic memory model.

Figure 5.6: The pipeline to build the hierarchical relational space

5.3.1 Correlations for Relations Between Elements

To analyze the relations between memory elements, we build a hierarchical relational space by a pipeline that contains two main processes: embedding and clustering. Firstly, the embedding maps the memory elements into a dense vector space. Then the clustering groups memory elements of the same level into multiple clusters.

We apply the embedding and clustering processes alternatively to build the event space and the episode space. The event space represents attributes and events as vectors, and the episode space represents event clusters and episodes as vectors. The relations between elements of the same levels or across different levels can be simply computed by their euclidean distances in the two spaces. The pipeline is shown by Figure 5.6.

There are plenty of choices for the algorithms of embedding and clustering. The embedding algorithm could be some adapted methods from natural language processing (NLP), such as the count-based method like the latent semantic analysis (LSA) [28] or the predictive method like the Word2Vec [90]. The clustering algorithm could be K-means[82], K-modoids [103], Affinity Propagation (AP) [42] and many others. In this work, we implement the LSA for embedding and the Kmeans for clustering.

The LSA method is originally designed for document retrieval by analyzing the word-document associations. Previous applications of the LSA method to memory are mainly aimed at getting semantic similarities for words [62][119]. However, there are significant differences between the LSA and our episodic memory: (1) The LSA has two levels: words

Figure 5.7: The first two dimensions of the event space in the UT2004 domain. The blue points and green circles represent parts of attributes and event clusters respectively

and documents, while episodic memory has three levels: attributes, events and episodes. The connections of the two spaces between the three levels need to be considered. (2) The LSA only deals with discrete words, while episode memory deals with both discrete and continuous attributes. (3) The LSA takes no consideration of order of words, while episodic memory considers order of events.

We adopt the LSA to analyze the relations between memory elements for the following reasons: (1) The method can map two levels of memory elements (for example, attributes and events) to the same space so that we can easily compute similarities of memory elements by their Euclidean distances. (2) It easily maps new episodes or cues into the built spaces by weighted linear combinations. (3) The method allows us to visually display ele-

Figure 5.8: The first two dimensions of the episode space in the UT2004 domain. The green circles and red crosses represent parts of event clusters and episode clusters respectively

ments in a 2D picture where similar elements are placed adjacent to each other, as shown in Figure 5.7 and Figure 5.8. (4) Many algorithms have been developed for efficient computation of LSA.

In addition, we borrow the idea of clustering of events and episodes from the EM-ART [135]. We adopt the Kmeans method for clustering due to its low computational complexity and easy control for the number of clusters. The event clustering is aimed at making generalization of events. Taking the implementation in the UT2004 domain as an example, the clustering picks out 200 typical event clusters to represent the total 17,579 events. This generalization can both diminish states of events to a much smaller scale as well as improve noise resistance of episodic memory model. The event clusters also take effects in connecting the event space and the episode space. Besides the purpose of making generalization, the episode clustering constructs a tree-like indexing for episodes that can

improve efficiency of retrieval. Given retrieval cues, the memory model can first compare with every cluster, and then go further only with episodes or events in relevant clusters.

The first two dimensions of the built event space are shown in Figure 5.7, in which the blue dots represent attributes and the green circles represent event clusters. Meanwhile, Figure 5.8 displays the first two dimensions of the built episode space, in which the green circles are also the event clusters and the red crosses are episode clusters.

The mathematical details of the implementation of the above pipeline are below.

(1) Build Attribute-Event Matrix Denote M_1 as the matrix of attributes and events. To build M_1 , events spread as vectors of all attribute values. The special cases are attributes that have continuous values. They are fuzzified so that the attribute values can be represented as linear combinations of finite fuzzy labels. The event vectors are constituted by the occurrences of all values of discrete attributes and all fuzzy labels of continuous variables. The entries with presence of discrete variable values are set as 1, those with absence of discrete attribute values are set as 0, and those with presence of fuzzy labels of continuous variables are set as their memberships between 0 and 1. As a result, all events share the same dimension as the length of total attribute values.

(2) Build Event Space We use singular value decomposition (SVD) method to decompose matrix M_1 into multiplication of three matrices $\bar{S}_1$, $\bar{V}_1$, and $\bar{D}_1$:

$$M_1 = \bar{S}_1 \bar{V}_1 \bar{D}_1',$$

where $\bar{S}_1$ and $\bar{D}_1$ are orthogonal matrices, and $\bar{V}_1$ is a diagonal matrix. The diagonal entries of $\bar{V}_1$ are called singular values which are arranged in a decreasing order. Each singular value corresponds to an eigenvector of matrix $M_1 M_1'$. They represent the main attribute-event relations weighted by the magnitude of the singular values. The **event space** is built by expanding these eigenvectors. To get main com-

ponents and reduce dimensions, we approximately represent M_1 by d_1 -dimension sub-matrices of the three multipliers. d_1 is the smallest number that makes the proportion between sum of the selected singular values and the total sum larger than a threshold th_1 . As a result, the matrix is approximately represented as:

$$M_1 \approx \widehat{M}_1 = S_1 V_1 D_1', \quad (5.1)$$

where S_1 and D_1 are the sub-matrices that keep the first d_1 columns of $\bar{S}_1$ and $\bar{D}_1$, V_1 keeps the first d_1 columns and rows of $\bar{V}_1$, and $\widehat{M}_1$ is the best rank- d_1 approximation to the matrix M_1 in the least-square sense. Attributes and events are vectors with the same dimension in the event space. The coordinates of attributes or events are the rows of matrix $C_1^{At} = S_1 V_1$ or $C_1^{Ev} = D_1 V_1$ respectively.

(3) Cluster Events To decrease number of event states, events are combined into k_1 clusters by the Kmeans method.

(4) Build Event Clusters-Episodes Matrix Denote M_2 as the matrix of event clusters and episodes. With the results of the preceding step, we build M_2 . Each episode spreads as a vector with dimensions of the total number of event clusters. The entries represent e occurrences of event clusters in episodes.

(5) Build Episode Space Applying SVD to matrix M_2 and performing an approximate decomposition with dimension d_2 , we have

$$M_2 \approx \widehat{M}_2 = S_2 V_2 D_2'. \quad (5.2)$$

The coordinates of event clusters or episodes are the rows of matrix $C_2^{Ev} = S_2 V_2$ or $C_2^{Ep} = D_2 V_2$ respectively.

(6) Cluster Episodes To improve retrieval efficiency, episodes are separated into k_2 clusters by the Kmeans method.

Figure 5.9: The retrieval by similarity with relations

Based on our correlation analysis for the relations between memory elements, we propose similarity measures with relations for cues and events or episodes. The retrievals of events and episodes based on their similarities with relations are illustrated in Figure 5.9. After embedding retrieval cues into the event space following the above pipeline, we can first retrieve relevant event clusters by their euclidean distances with cues in the event space. Then we retrieve relevant events from those retrieved clusters. For episode retrieval, the computation needs to go further. After embedding events of cues into the event space, we continue to embed the corresponding event clusters of cues to the episode space. Similarly, we first retrieve relevant clusters of episodes by their euclidean distances with the cues in the episode space and finally retrieve the most relevant episodes.

5.3.2 Correlations for Weights of Attributes

The proposed mixed correlation analysis also considers the weights of attributes in different episodes. Besides assigning weights based on their occurrence frequencies in the EM-ART model [135], we also analyze the specialness of attributes at particular episodes. Therefore, we adopt the TF-IDF indexing [110], which is an existing technique for document retrieval, to compute weights of attributes in episodes. The TF-IDF indexing assigns each attribute

a weight to indicate the extent that it can represent the whole episode. The TF value computes the frequency that an attribute appears in one episode, and the IDF value, indicating specialness of attributes, computes how common this attribute appear in all episodes. Thus attributes have high weights in some episodes if they frequently appear in these episodes while rarely appear in the others.

Denote the collection of episodes is EP , the total number of episodes is N , the total number of episodes that contains an arbitrary attribute a is N_a , the frequency that the attribute a appears in an episode ep is $f_{a,ep}$ and the frequency that it appears in a cue c is $f_{a,c}$. Then the weight of the attribute a at the episode ep is

$$w_{a,ep} = tf(a, ep) \cdot idf(a, EP) = (1 + \log f_{a,ep}) \cdot \log \frac{N}{N_a}. \quad (5.3)$$

and the weight of the attribute a at the cue c is similar:

$$w_{a,c} = tf(a, c) \cdot idf(a, EP) = (1 + \log f_{a,c}) \cdot \log \frac{N}{N_a}. \quad (5.4)$$

Thus we can represent the ep and c as vectors V_{ep} and V_c which have the same length as the total of attributes. The entries of the vectors are the weights of the attributes. Because the TF-IDF works well only if candidates have enough attributes for statistical concern, we take weights of attributes in events the same as they are in particular episodes that contain the events.

Based on the correlation analysis for weights of attributes, the similarity with weights for the episode ep and the cue c can be computed by the dot product of the two normalized vectors:

$$S_w(c, ep) = V_{ep} \cdot V_c / \|V_{ep}\| \|V_c\| \quad (5.5)$$

The computation of the similarity with weights for events and cues is similar. The most relevant events or episodes are the ones with the maximum similarity score.

Figure 5.10: The sliding window method to analyze event-order correlations of episodes

5.3.3 Correlations for Order of Events

The order of events is critical for episode retrieval, especially given incomplete and noisy cues. Our work analyzes the event-order correlations of episodes by a sliding-window method based on the edit-distance algorithm [78] that was originally designed for string matching.

The correlation analysis for the order of events in episodes is aimed at finding the best partial sequence matches between episodes since it is common that episodes have different lengths. To analyze the event-order correlation of two episodes, we compare the shorter episode with all subsequences of the longer episode, where the subsequences have the same length with the shorter episode. The correlations of the two episodes is determined by the smallest operations to turn the shorter episode to one of those subsequences.

Based on the analysis of the event-order correlation, we propose similarity measure for event-order of cues and episodes. The pseudo-codes are described in the Algorithm 1. The inputs are arbitrary cues and episodes. The event-order similarity is computed by one minus the normalized edit-distance, which a value between 0 and 1. The similarity is 1 if the two sequences are exactly the same. If the cue is shorter than the episode, we use the function “getSubSequences” to get all the subsequences from the episode with the same length of the cue. We have two choices for subsequence selection. The first method selects all the

Algorithm 1: Compute event-order similarities between cues and episodes

```

1: Def getEventOrderSimilarity(cue,episode,w=1)
2:   m=len(cue), n=len(episode)
3:   IF  $m \geq n$ 
4:     RETURN 1-editDistance(cue,episode)/m
5:   ELSE
6:     Sims=list()
7:     FOR ep_seq in getSubSequences(episode,m,w)
8:       sim=1-editDistance(cue,ep_seq)/m
9:       Sims.append(sim)
10:    END FOR
11:    RETURN max(Sims)
    
```

combinations with length m , thus the overall complexity of the algorithm is $C_n^m O(mn)$. While the second method only selects the continuous subsequences within a sliding window to move w steps once, as shown in Figure 5.10. It has smaller complexity as $(m - n + 2 - w)O(mn)$. The computation speed will increase and the accuracy will decrease with larger w . In this chapter, we implement the second choice to get the continuous subsequences of episodes since the cue events are usually given in continuous order. Finally, the algorithm returns the maximum of all the event-order similarities of the candidate subsequences.

Because the computation of the edit-distance is relatively slow for long cues and episodes, the sliding step of the window w should be carefully selected to improve the computation efficiency without significant drop of the results as well. Instead of being a constant, w should get small if the length difference of the cue and the episode $m - n$ is small, while get large on the contrary. However the performance will drop for insufficient comparison if w gets larger than the length of the cue n , especially in the cases with large comparative length difference ($m - n$ is large but n is small). Therefore, the step of the window is set below:

$$w = \min[(m - n)/W + 1, n]. \quad (5.6)$$

W is integer constant set by clients and $/$ represents the exact division with no reminder. So we get the number of computation to be $\max[W, m/n - 1]$ and the total complexity to be $\max[W, m/n - 1]O(mn)$. In another words, the constant W limits the times of computation

Figure 5.11: The combination of algorithms for aspects of similarity measure

in most cases with small comparative length difference of cues and episodes.

5.3.4 Overall Similarity Measure

Based on the correlation analysis of memory elements and the similarity measures for each type of correlation, we propose a novel similarity for cues with events or episodes.

The cues for event retrievals are collections of attributes and those for episode retrievals are sequence of events. The retrievals of events and episodes are similar in considering the content similarity but different in that episode retrievals need to consider the order similarity as well.

As illustrated in Figure 5.11, we compute the content similarities of the cues with the candidate events or episodes. Firstly, we compute the similarities with weights S_w and the similarities with relations S_r and then get the rankings R_w and R_r of all the candidates respectively. The computation of S_w has been illustrated by the equation 5.5 and the computation of S_r has been introduced by Figure 5.9. The content similarities S_c are computed by the sum of the reciprocals of the two rankings:

$$S_c = (1/R_w + 1/R_r)/2. \quad (5.7)$$

Then the content similarities range from 0 to 1. The reciprocals of the two rankings are

adopted so that the desirable event can get good ranking if at least one of the two rankings is high. The two algorithms have different suitable cases: the similarities with weights work better for the cases of small noises while the similarities with relations have better performance for those with big noises. Therefore, the equation 5.7 takes the benefits of the two algorithms. Afterwards, we return the candidate with the maximum content similarity.

The retrievals of episodes need to take further consideration of the order of events. Because the computation of the similarities with event-order is time consuming, we just select the best K ($K=20$ for our setting) candidates by the content similarities for event-order matching as described in the Algorithm 1. The order similarities S_o are computed based on the sequence of the event clusters of the K candidates and the cue. Using the event clusters instead of the events can improve the noise resistance of the model.

Finally, we get the overall similarities S for the K candidates by the production of the content similarities and the order similarities:

$$S = S_c \cdot S_o. \quad (5.8)$$

The production is adopted so that the algorithm will only the candidates that have good matches on both the contents and the event-orders with the cues. The candidate episode with the highest overall similarity will be returned.

5.3.5 Proposed Episodic Memory Model

The proposed episodic memory model contains the training, encoding, storage and retrieval of memory data (the forgetting is not considered here). Figure 5.12 displays the overall structure of the proposed MCAEM model. The procedures are deliberated below.

(1) Training Build the weights of attributes and the hierarchical relational space for all the episodes in the datasets.

(2) Encoding New episodes get the weights of their attributes by the equation 5.3 and

Figure 5.12: The framework of episodic memory

are embedded into the hierarchical relational space following the pipeline shown in Figure 5.6.

(3) Storage The storage mechanism has three hierarchies: short-term memory (STM) or working memory (WM), long-term memory (LTM) and hardware storage. Figure 5.13 shows the detailed workflow. The STM has limited space to store fixed episodes, while the LTM has much larger storage for all historical episodes organized by the cluster-based indexes. The functionality of STM is to accelerate the retrieval process. A simple implementation of STM could be a queue with the first-in-first-out (FIFO) principle. The inputs to STM are the new encoded episodes (which are also stored in LTM with indexes) and the retrieved ones from LTM. If the STM is full, the bottom episode will be deleted. Nevertheless, if one STM episode is retrieved, it will go to the top of the STM queue again. At the beginning or the end of the memory process, the LTM will load from or store into the hardware.

(4) Retrieval The retrieval follows the order from STM to LTM. We first compares the episodes or events in the STM with the given cues. The similarities of the episodes or events and the cues are computed based on the proposed algorithm in the sec-

Figure 5.13: The mechanism of short-term and long-term memory

tion 5.3.4. Then we return the episodes or events with similarities above the predefined threshold. If no episode or event can be returned, we go further for the LTM.

5.4 Application on Intelligent Agents

In this section, we evaluate effects of the proposed MCAEM model on improving agent performance in a first-person shooter game, Unreal Tournament 2004 (UT2004). Agents in UT2004 game are controlled by a Java platform Pogamut 3 [48]. The proposed model is compared with the EM-ART model [135] in experiments below.

5.4.1 Specific Adaptations for Intelligent Agents

All classes of attributes considered in the UT2004 domain are displayed in Table 5.2. To analyze mixed correlations of memory elements, all attributes in these classes need to be represented as discrete states. The special cases are the *attribute classes* in continuous value domains (the classes with the double data type such as “reward”, “weapon-Reward”, etc. in Table 5.2). Instead of using a crisp discretization which separates the

continuous domains into several crisp sets with binary memberships, we adopt a fuzzy discretization that separates them into fuzzy sets with continuous memberships. For example, Figure 5.14 shows the difference of a fuzzy discretization and a crisp one for the class “enemyDistance”. Figure 5.14 indicates that six fuzzy attributes are considered in this class. Given an input “enemyDistance=345”, the result of the crisp discretization is “enemyDistance=EXTREME_NEAR” while the result of the fuzzy discretization is “enemyDistance=0.48*EXTREME_NEAR + 0.52*VERY_NEAR”.

Figure 5.14: The membership functions for enemy distance

To evaluate our model, we conduct experiments on a UT2004 dataset and the platform.

Table 5.2: All classes of attributes considered in the UT2004 domain

String subject; String predicate; String directObject; String indirectObject; String location; double reward; double weaponReward; String preAction; String curAction; String agentWeapon; String enemyWeapon; double curHealth; boolean beingDamaged; boolean enemySeen; double enemyDistance; boolean reachableWeapon; boolean reachableHealthPack; boolean reachableAmmo;
--

5.4.2 Test on UT2004 Dataset

To test performance of the proposed episodic memory model directly, we conduct several experiments on a dataset generated in the UT2004 game.

Experiment Setup

The dataset contains 394 episodes of battles, 17579 events and 240 distinctive attributes in total. The experiments evaluate retrieval accuracies of episode retrievals given two types erroneous cues: noisy cues and partial cues. The generation of the two types of cues is deliberated below:

(1) Noisy cue Noisy cues are generated by adapting attributes in randomly selected events or episodes from the dataset with noise rate $\sigma_n < 1$. For attributes in classes with discrete value domains in Table 5.2, one attribute has σ_n probability to be replaced by another one in the same class. For attributes in classes with continuous value domains, however, an attribute value v will be adapted by a random error from an interval $[-\sigma_n v, \sigma_n v]$. By default, $\sigma_n = 30\%$.

(2) Partial cue Partial cues are generated by taking parts of original events or episodes. The partial rate $\sigma_p < 1$ denotes the ratio of contents to be reserved. In addition, we only continuous event subsequences of episodes in these experiments. By default, $\sigma_p = 30\%$.

Table 5.3: The means and standard errors of episode retrievals of the MCAEM model and EM-ART model

Models	Noisy Cue	Partial Cue
MCAEM	M=0.990 SD= 6.4×10^{-5}	M=0.979 SD= 1.6×10^{-4}
EM-ART	M=0.801 SD= 9.3×10^{-4}	M=0.898 SD= 7.6×10^{-4}

(a) Noisy cues

(b) Partial cues

Figure 5.15: Comparing episode retrieval of the MCAEM model and EM-ART model on the UT2004 dataset for twenty trials. The blue and red curves represent the MCAEM model and EM-ART model respectively

Results

We compare performance of episode retrievals of the proposed MCAEM model and the EM-ART model on the UT2004 dataset for 20 trials given the two types of erroneous cues. In each trial, the retrieval accuracy is computed by the average performance over 100 randomly generated cues. Figure 5.15 displays results of the MCAEM model and EM-ART model represented by blue and red curves respectively. The means and standard errors of the results are displayed in Table 5.3. By a two-pair T-test, the proposed MCAEM model significantly outperforms the EM-ART model on episode retrievals given noise cues ($t = 29.7, p = 2.2 \times 10^{-17}$) and partial cues ($t = 11.6, p = 4.4 \times 10^{-10}$).

Discussion

The results of above experiments indicate that our MCAEM model has better error resistance than the EM-ART model. The better performance in noisy cues results from the consideration of the relations between memory elements. In the case that an attribute in the classes with discrete value domains are substituted by a similar one in the same class, the relation between these attributes help the MCAEM model retrieve correct source episodes. In the case that attributes in the classes with continuous value domains are disturbed by noises, the fuzzy discretization described in the section 5.4.1 is more robust than the crisp discretization.

The better performance of the MCAEM model given partial cues is mainly because of the proposed event-order similarity measure. With the analysis of relations of elements and weights of attributes to filter out irrelevant episodes by content correlations, the remained episodes are finally ranked by their event-orders. The better results demonstrate that the proposed event-order algorithm is more robust for event positions.

5.4.3 Test on UT2004 Platform

In this section, we evaluate the proposed MCAEM model by agent performance on the UT2004 platform. The MCAEM provides three cognitive capabilities: item finding, weapon selection and decision making. We make comparisons of the agents with all combinations of the three capabilities as well as the default agent and the agent with the EM-ART model on the platform.

Experiment Setup

Experiments are conducted in the game mode called “Deathmatch” within the map “DM-1on1-Albatross”. In the “Deathmatch” mode, two agents run around the battlefield trying to kill each other. Agents can collect weapons and ammo to enhance shooting power, and collect medical packages to improve health level. The collected items will respawn period-

Table 5.4: The learned locations for a few items

Item	Location	Item	Location
SniperRifle	<i>InventorySpot356</i>	ShockAmmo	<i>InventorySpot345</i>
RocketLauncher	<i>InventorySpot357</i>	MinigunAmmo	<i>InventorySpot338</i>
Minigun	<i>InventorySpot355</i>	MiniHealthPack	<i>InventorySpot319</i>
FlakCannon	<i>InventorySpot354</i>	HealthPack	<i>InventorySpot362</i>

Table 5.5: The retrieved weapon effects with given distances

Distance	1000	1500	2500	3000
RocketLauncher	68.37	71.51	69.21	70.41
SniperRifle	54.97	56.52	54.55	59.81
BioRifle	39.10	36.26	38.29	38.46
ShockRifle	36.38	38.43	33.61	34.92
LinkGun	29.36	25.53	27.06	26.13
FlakCannon	22.96	16.45	19.23	16.83
Minigun	8.30	8.56	8.64	8.64
AssaultRifle	6.87	7.12	6.95	7.08

ically at fixed locations. Agents are killed if their health values come to zero.

The default agent called *HunterBot* makes decisions according to some predefined rules with certain randomness. It has the following available actions:

- (1) **Engage:** An agent approaches and shoots at an enemy when the enemy is visible.
- (2) **Escape:** An agent in a battle escapes from an enemy to a safer place.
- (3) **Pursue:** An agent pursues an enemy who becomes invisible.
- (4) **Collect Items:** An agent walks around and only picks up visible items.
- (5) **Collect Weapons:** An agent walks around and only picks up visible weapons and ammo.
- (6) **Collect Medicines:** An agent walks around and only picks up visible medicines.
- (7) **Random Walk:** An agent walks around and picks up items randomly.

The proposed episodic memory model is off-line trained by 100 battles of two *HunterBots* (one battle is counted if an agent kills or is killed by an enemy one time). The attributes

shown in Table 5.2 are considered in these experiments. The proposed model supports three cognitive capabilities: item finding, weapon selection and decision making. The first two capabilities are the applications of relations of attributes, and the third one is the application of episode retrieval. They are described below:

(1) Item Finding Find the most possible locations given an arbitrary item (a weapon or medicine . . .). The pseudo-codes are shown in Algorithm 2. The function “getAllLocationAttributes()” returns all attributes in the class of ”location”. The algorithm gets the most related locations to a given item. The item finding enables agents to collect items which have been met before but are currently invisible. For example, an agent can go directly to the best collectible weapon at the very beginning, which largely improves its probability of survival. Some example learned locations for a few items are shown in Table 5.4.

(2) Weapon Selection Select the available weapon with the best expected reward given a distance from a visible enemy. The pseudo-codes are shown in Algorithm 3. The weapon selection enables agents to select the best available weapons to shoot at an enemy. The input “distance” is a continuous value that is first fuzzified into a combination of several fuzzy “enemyDistance” attributes as shown in Figure 5.14. The function “getAllAvaliableWeaponAttributes()” returns all weapons in the class “IndirObject”. The class “weaponReward” contains several fuzzy attributes (“small”, “medium”, “large”, “very large” and “extreme large”). The “reward” is computed by the process of defuzzification. Some example defuzzified weapon rewards given distances are shown in Table 5.5, which demonstrates that (1) effects of weapons change with distance. (2) some ranks of weapons are influenced by distance, for example, the “BioRifle” and the “ShockRifle” exchanges their ranks given the distance 1000 and 1500.

(3) Decision Making Find the action with the best expected reward given all other attributes in the current context. The rewards of actions are set below: the reward

Algorithm 2: Algorithm to find items

```

1: Def itemFinding(item)
2:   Get the coordinates C of the attribute "directObject=item".
3:   relevances={ }
4:   FOR location in getAllLocationAttributes()
5:     rel=cosine(C, location.coordinates).
6:     relevances[location]=rel
7:   END FOR
8:   RETURN the location with maximum rel

```

is 0.5 if agents pick up items or hurt an enemy; the reward is -0.5 if agents get hurt, and the reward is 10 or -10 if agents kill or get killed by an enemy. The pseudo-codes are shown in Algorithm 4. We describe the algorithm as follows: For each possible action, we combine it with other attributes in the current context as a cue. The cue is provided to episodic memory to retrieve the most related episodes. In each episode, we first find all events, if any, that share with the cue the same "preAction", "curAction" and "enemySeen" attributes. Discounted rewards are computed from these events to the end of the episode with a discounting factor 0.8. An expected reward of this action is computed by an average of all discounted rewards. Finally, the action with the best expected reward is returned. The decision making enables agents to make decisions based on lessons learned from past experiences.

To evaluate the MCAEM model, we integrate default *HunterBot* agents with one to three above cognitive capabilities (seven combinations). Agents integrate some default capabilities instead for missing ones. The default item finding makes agents collect only visible items as well as random walking. The default weapon selection randomly selects available weapons. And the default decision making keeps the predefined decisions in the *HunterBot* agent.

In all, we have seven distinct MCAEM-based agents and a default *HunterBot* agent. These agents will combat with an agent that integrates the EM-ART model [136, 135]. The EM-ART model also supports cognitive capabilities of weapon selection and decision making. Different with our work, Wang et.al [136] adopts additional ART neural networks

Algorithm 3: Algorithm to find the best weapon

```

1: Def weaponSelection(distance)
2:   Get fuzzy representation of the distance by the combination of the
   “enemyDistance” attributes.
3:   rewards={}
4:   FOR weapon in getAllAvaliableWeaponAttributes()
5:     context=weapon + distance
6:     relevance={}
7:     FOR weaponReward in getAllWeaponRewardAttributes()
8:       rel=cosine(weaponReward.coordinates,context.coordinates)
9:       relevance[weaponReward]=rel.
10:    END FOR
11:    Get reward by defuzzifying the relevances to every weaponReward attribute.
12:    rewards[weapon]=reward
13:  END FOR
14:  RETURN the weapon with maximum reward.

```

for the two capabilities. The episodic memory in the EM-ART is used to filter redundant episodes for the two neural networks. In addition, rewards of actions are learned by a Q-learning algorithm. In these experiments, the EM-ART-based agent trained by 1800 battles, called *EMART-1800*, is adopted.

Results

All tested agents in our experiments play against the *EMART-1800* agent for 30 matches. In one match, agents have 10 lives and will respawn in random locations immediately after death. The winner is the first agent that has killed the enemy ten times. The agent performance measured by a **Kill/Killed** rate, a ratio of the number of killing against the number of being killed. The default rate is 1 (for two same agents). Higher rates refer to better performance.

The results of experiments are shown in Figure 5.16. The red line is the baseline **Kill/Killed** rate of the *EMART-1800* agent. We have following observations from Figure 5.16

- (1) The default *HunterBot* agent has worse performance than the baseline as well as that of other agents, thus the proposed MCAEM model and the EM-ART model have

Algorithm 4: Algorithm to make decisions

```

1: Def decisionMaking(context)
2:   rewards={}
3:   FOR action in getAllAvaliableActionAttributes()
4:     cue=context+“currentAction=action”
5:     filter = ”previousAction” and ”enemySeen” in the context
6:     candidates = getAllRelatedEpisodes(cue) filtered by filter
7:     reward = 0
8:     FOR episode in candidates
9:       FOR event in episode filtered by filter
10:        Get the discounted reward r from event to the end of episode
11:        reward += r
12:      END FOR
13:    END FOR
14:    reward = reward / sizeof(candidates)
15:    rewards[action] = reward
16:  END FOR
17:  RETURN action with maximum reward

```

improved agent performance.

- (2) Agents that integrate the MCAEM model have better performance than the baseline, so the MCAEM model outperforms the EM-ART model in improving agent performance even though the MCAEM model has much smaller training dataset.
- (3) The weapon selection provided by the MCAEM model has better performance effect than the capabilities of decision making and item finding.
- (4) The capabilities of decision making and item finding have limited improvement of agent performance when they are combined with weapon selection.

5.4.4 Discussion

The results show that the proposed MCAEM model has significantly improved agent performance with much less training cases compared to the EM-ART model. This is because the MCAEM makes better use of information in training cases by considering the attribute-attribute and event-episode relations. The results indicate that these considerations support

Figure 5.16: The kill/killed rates of tested agents against the *EMART-1800* agent

more accurate retrievals of attributes and episodes that support better cognitive capabilities for agents. For example, exploring relations of weapons, distances and weapon effects in the event space can improve weapon selections. And accurately retrieved relevant episodes can improve the computation of expected rewards of possible actions in current contexts.

In addition, it is easier for the MCAEM model to complete relation-based tasks such as item finding and weapon selection. This is because the MCAEM model provides task-irrelevant preparations for relations between elements. The preparations include convenient folding-in of arbitrary combinations of attributes and easy similarity computation by euclidean distances or cosines of vectors. On the contrary, the EM-ART model [135] needs to train a new semantic neural network for each relation-based task from scratch, which is tedious and time-consuming.

Thirdly, the decision making of the EM-ART model uses reinforcement learning, in which the EM-ART model works as a filter by selecting key episodes and events. However the proposed MCAEM model contributes directly to decision making by analyzing discounted rewards of previous actions in similar contexts based on retrieved episodes.

The results of experiments indicate that reinforcement learning probably has fallacy in over generalization for states and rewards in complicated contexts. The contexts are complicated in the meaning that elements in contexts are massively shared by different tasks and purposes, thus decision making algorithms should collect relevant experience for certain tasks. However, reinforcement learning stores no specific memory but overall impression for past experiences, relevant or not relevant, to make decisions for new cases. Nevertheless, our episodic memory can improve decision making by considering only the most relevant experiences to current contexts. The results indicate that the episodic-memory based decision making probably has advantages in complicated contexts than the reinforcement learning.

Finally, among the three capabilities supported by the MCAEM model, the weapon selection has better performance improvement than the decision making and item finding. This is because weapon selection has the most direct impact on killing the enemy. The item finding has the worst effect for two reasons. Firstly, the battle field is so small that the default item finding that collects visible items with random walk can still have a not bad effect. Secondly, good weapons of agents will be taken away by the enemy who kills them. Thus sometimes agents without the advanced item finding can get a good weapon quickly from their enemies.

5.5 Application on Social Companion

Some necessary adaptations of the MCAEM model need to be considered for social companion-user interactions. Compared to intelligent agents, the application on social companions has different attributes and training process. The consideration of words from spoken languages results in a large-scale attribute domain, which is more challenging for episodic memory models than small domains of AI. In addition, it is more difficult to collect a large amount of episodes from interactions than from AI domains, which forces us to adjust our training process.

Figure 5.17: Event structure with attributes for social companion-user interactions

5.5.1 Specific Adaptations for Social Companions

For the application of dialog-based companion-user interaction, the event structure with attributes is shown in Figure 5.17. The attribute classes take into consideration known or unknown words in user utterances and other context information, such as subjects and objects of speech, current emotions and moods of companions and social time, etc. Obviously, the attribute classes can be further extended to include more concepts such as relations with users and social roles according to applications. To extract attributes from natural language inputs, a natural language processing tool NLTK* is utilized to fulfill tasks of tokenization, stemming, tagging, parsing and semantic reasoning. Known words are those contained in the dictionary, while unknown words, such as some proper nouns and human names, need to be further learned in dialog contexts. Users first get registered by providing their names and face images to the face recognition module. Thus our social companion can get their names immediately once users are recognized. In this application, each interaction is regarded as an episode, in which events correspond to every turn-taking in dialog. The subjects and objects of events are determined as follows: When a user is speaking, the subject is by default the user while the object is our social companion. On the contrary, when the companion is speaking, the subject is the companion while the object is the user. Emotions and moods are regarded as important influencing factors to human memories [12]. In this work, the emotion class contains a neutral state and six OCC emotions introduced in Section 3.2. The six emotions are “Happy-for”, “Pity”, “Gratitude”, “Anger”, “Joy” and “Distress”. The mood

*<http://www.nltk.org/>

class contains 8 attributes corresponding to the 8 octants of the pleasure-arousal dominance (PAD) space [89], shown in Table 5.6. Because human brain remembers time and date information in a social way, we denote social time as “morning”, “afternoon” and “night”, and denote social date from “Monday” to “Sunday”. In the interaction, we also adopt social concepts like “Yesterday” or “Tomorrow” instead of actual dates on a computer screen. Undoubtedly, this rough classification can be further refined based on user needs.

Table 5.6: Mehrabian mood types.

Octants	Mood Types	Octants	Mood Types
P+, A+, D+	Exuberant	P+,A-,D+	Relaxed
P-, A-, D-	Bored	P-,A+,D-	Anxious
P+, A+, D-	Dependent	P+,A-,D-	Docile
P-,A-,D+	Disdainful	P-,A+,D+	Hostile

The first embedding step of the event space is adjusted for the difficulty in collecting a large amount of episodes from interactions. Instead of training one vector space for all attributes, we train two spaces for context attributes and word attributes respectively. The word attribute space is trained based on a latest Wikipedia English dataset. The space represents words as vectors in 400 dimensions, and similarities of words can be easily computed by Euclidean distances or cosines. On the other hand, to build a context attribute space (which has a limited scale), 200 dialogs with thousand sentences between users and an AIML 2.0 chatbot are collected. Although the AIML chatbot has limited ability to keep conversation over same topics, participants are required to shift topics if they get nonsense answers from the chatbot. Events are encoded as the combination of the word embeddings and the context embeddings. The event similarity is calculated by the average of the word similarity and the context similarity. Although this adapted event space has disadvantage that we cannot get direct relations of word attributes and context attributes, it guarantees that all attribute embeddings are fully trained.

Figure 3.3 shows the framework of our social companions. The episodic memory model provides supplemental memory functionalities such as retrieval of similar utterances, con-

textual features or meta-information. It endows the companion with awareness of past experiences, and improved dialog performance.

5.5.2 Memory-Related Capabilities

In this section, we discuss how the proposed episodic memory endows companions with some capabilities. The capabilities will improve user experience in dialog-based interactions with social companions.

Retrieving Previous Utterances

The proposed episodic memory allows social companions to recall previous utterances similar to the current context. The utterance retrieval is based on not only users' inputs, but also other context attributes as shown in Figure 5.17. A retrieval happens either spontaneously during dialog or deliberately to retrieve expected answers for users' questions. For example, if a user describes his experience ("Yesterday, I played basketball with my friends") or asks question ("Do you remember my hobby"), our companion will recall similar utterances from this user or from other users with simple paraphrasing ("I remember you said that your hobby is playing basketball. In addition, Tom also said he likes to play basketball"). Users usually show great interest if their friends' names are mentioned by the companion. The mixed correlation analysis in our episodic memory model allows users to speak incomplete sentences with free selection of words. Algorithm 5 describes the retrieval process in details.

Retrieving Related Attributes

The memory model also recalls other attributes similar to the context. For example, once an event is retrieved, the related emotions and moods are also retrieved. The retrieved emotions and moods will influence the affective states of the companion. Gomes *et al* [51] proposed a similar idea but they only consider relations between emotions and locations. Our work

Algorithm 5: Retrieve utterances similar to context

```

1: Input current context  $C$ , episode threshold  $ep\_th$ , event threshold  $ev\_th$ 
2: Select episodes with similarity  $C$  larger than  $ep\_th$ 
3: Set  $candidates = [ ]$ 
4: FOR each event  $e$  in the selected episodes
5:   IF similarity  $sim(e, C)$  is larger than  $ev\_th$ 
6:      $candidates.append(e)$ 
7: END FOR
8: WHILE  $len(candidates) > 0$ 
9:   Randomly select event  $ev$  from  $candidates$  with probabilities proportional to the
   similarities
10:  Set answer  $ans = None$ 
11:  IF sentence  $ev.sent$  is question
12:    Get the next event  $ev'$  in the same episode
13:     $ans = ev'.sent$ 
14:  ELSE
15:     $ans = ev.sent$ 
16:  IF  $ans$  is not in recent  $n$  retrievals
17:    RETURN  $ans$ 
18: END WHILE
19: RETURN  $None$ 

```

allows mutual recalls for arbitrary two attributes. The model also supports computation of an emotional relationship with users by averaging relations between the users and the six emotions.

Retrieving Meta-Information

Retrievals of meta-information of episodes such as time tags or interacted users are also required. In overall, the achieved capabilities are listed below: (1) to check whether a specific user is known or unknown; (2) to report whether a specific user was met on some date, report all the users met on a certain date, or report the latest date when the companion met a specific user; (3) to report recent activities of the companion; (4) to give a summary of the last conversation with a specific user; and (5) to give information about a specific user. In the current implementation, we just select and paraphrase a specific user replies to the questions from the companion.

5.5.3 User Study

To explore our episodic memory model in companion-user interactions, a within-subject user study is conducted by integrating our memory model with our virtual human Sophie. Three hypotheses are proposed for the user study:

Hypothesis 1 *The proposed episodic memory can significantly improve the retrieval performance in companion-human interaction.*

Hypothesis 2 *The existence of the memory model can significantly improve the user experience in various aspects.*

Hypothesis 3 *The improved aspects of user experience, if any, have significant correlations with the subject-perceived memory performance.*

Methodologies

A total of 20 participants (16 males and 4 females) were recruited for this study. The mean age of them was 31 ranging from 25 to 54. Participants had above average familiarity with virtual human and below average knowledge about episodic memory. The above information was collected by inquiry forms before the experiments. Figure 5.18 shows the example images captured in this study.

Participants were required to interact with our virtual human Sophie by free conversation in English. To correctly recognize their speech, participants were suggested to give short, slow, clear and continuous responses. Otherwise, they had to repeat their sentences if the speech recognition failed.

To evaluate the performance and benefits of our episodic memory, each participant was required to interact with **two versions of Sophie**, one with the memory component and one without, in random order. In this within-subject study, the existence of the episodic memory is the only independent variable. It is worth pointing out that Sophie without our memory component still has a certain ability of knowledge learning provided by the AIML

Figure 5.18: Some participants

2.0 chatbot. However, the chatbot is able to learn only if users speak in certain sentence formats, using a kind of information extraction (IE) technique. For example, “my hobby is playing basketball” can be learned, while “I like to play basketball” cannot.

Each interaction with Sophie was divided into two sessions: the **collection session** and the **retrieval session**. In the **collection session**, participants interacted with Sophie as new users. They first registered themselves in front of Kinect using the face recognition component so that Sophie could build connections between user faces and names. To get better understanding of them, Sophie asked questions about four topics: nationality, current work, research topic and hobby. Participants were required to give direct answers to these questions. The answers could be complete sentences or simple words. General answers (such as “I don’t know” or “I have no idea”) were also acceptable. After each question, Sophie further discussed with the participants about the same topic for a while. During the dialog, participants could ask Sophie any question at any time too (for example, “What’s your name”). After Sophie gathered all the wanted information, she would end the dialog.

In the **retrieval session**, participants met Sophie again several hours later. After greetings, participants were required to ask Sophie questions about their last conversations. For example, “What is my nationality?”, “Do you know my research topic?”. Besides the question forms, the word selections were also arbitrary. For instance, participants could ask their “job for now” or “study focus” instead of “current work” or “research topic”. With the permission of participants, Sophie would ask another four random questions so that Sophie could collect more information about them. Participants were allowed to end the interactions at any time by saying “goodbye”.

Evaluation Metric

After interactions with the two versions of Sophie, participants were required to complete a 7-Likert questionnaire with a total of 26 questions. The detailed questions are presented in the Appendix B. Several dependent variables were considered in these questions. Several dependent variables were considered in these questions, including the *Memory impression*, *User engagement*, *User responsiveness*, *Dialog appropriateness*, *Learning ability*, *Usefulness*, *Potential for long-term interaction*.

Memory impression measures the subject impression of participants for the retrieval performance of Sophie. Questions for *Memory impression* are displayed in Table 5.7. We borrow the concepts of *user engagement* and *user responsiveness* from [69]. *User engagement* refers to the extent that users get involved in the interaction and *user responsiveness* considers how much users make eye contacts, sounds or facial expressions in response to Sophie. The last four variables correspond to the four characteristics introduced for long-term dialog companions in [24]: the conversational ability, adaptability, utility and long-term interaction. *Dialog appropriateness* refers to the ability of companions to generate and understand spontaneous dialogs within social context. *Learning ability* requires companions to grow with experience and to adapt to user preferences. *Usefulness* describes the benefits of dialogs with companions to users in various tasks like searching information or memory assistance. Last but not the least, *potential for long-term interaction* considers user

willingness to continue interacting with the companion in future.

Besides these subjective measures, we objectively calculate the *Memory performance* by the percentage of the correctly answered questions by Sophie in the retrieval session.

Table 5.7: Questions for retrieval performance

1. I believe that Sophie will remember me when I meet her again.
2. I believe that Sophie talks to me in aware of our past conversation.
3. Sophie can remember almost everything I have talked.
4. Sophie has good memory of past dialogs.
5. Sophie can recall related knowledge from other people.
6. The recalled memories are coherent to current context.

Results

After collecting data from a user study, we utilized the two-pair T-test to analyze the dependent variables. The results are displayed in Figure 5.19, and the detailed scores are given in Table 5.8.

The first hypothesis predicted that Sophie with the proposed episodic memory would have better performance in retrieval. By the objective measure, this hypothesis is supported by the fact that Sophie with our memory module has significant higher retrieval accuracy than Sophie without ($t = 13.412, p < 3.87 \times 10^{-11}$). What's more, participants also express a better impression for the retrieval performance to Sophie with our memory module ($t = 7.633, p < 3.48 \times 10^{-9}$).

The second hypothesis predicted that companion-human interaction would be improved by our memory component. This hypothesis is partially accepted. Subjects find that interactions with the memory-equipped Sophie have significant better *dialog appropriateness* ($t = 2.884, p < 0.006$), *learning ability* ($t = 6.055, p < 4.79 \times 10^{-7}$) and *usefulness* ($t = 4.367, p < 9.36 \times 10^{-5}$). The participants also show stronger willingness to have further interactions with the memory-equipped Sophie in future ($t = 2.632, p < 0.012$). Although participants scored the memory-equipped Sophie higher in *user engagement* ($t = 1.978, p < 0.055$) and *user responsiveness* ($t = 1.216, p < 0.232$), no significant difference

Figure 5.19: Effects of the memory component

is found between the two Sophies with the lower bound $p = 0.05$ (the former is significant for the one-pair T-test).

The third Hypothesis predicted that the improved aspects of user experience are significantly correlated to the participants' subjective impression for the memory performance. We measure the correlations of the dependent variables with the subject-perceived memory performance (namely the *Memory impression*) by Pearson's r , of which significance is measured by a two-pair T-test. The correlation results are described in Table 5.9. The four significantly improved aspects relevant for the Hypothesis 2 also have significant correlation with the subject-perceived memory as listed below: (1) *learning ability* ($r = 0.89, p < 4 \times 10^{-6}$), (2) *usefulness* ($r = 0.81, p < 0.0001$) (3) *potential for long-term interaction* ($r = 0.73, p < 0.0001$), (4) *dialog appropriateness* ($r = 0.71, p < 0.002$). Meanwhile, the two aspects, *user engagement* ($r = 0.59, p < 0.017$) and *user responsiveness* ($r = 0.26, p < 0.338$), that are not significantly improved in this study have no significant relations with the subject-perceived memory.

Table 5.8: Results for objective measure and the subjective user study

Measures	With Memory	Without Memory	p
Memory performance	M=0.90, SD=0.02	M=0.12, SD=0.04	3.87×10^{-11}
Memory impression	M=5.98, SD=0.71	M=2.94, SD=1.63	3.48×10^{-9}
User engagement	M=4.75, SD=1.13	M=4.00, SD=1.26	0.055
User responsiveness	M=4.72, SD=1.13	M=4.29, SD=1.08	0.232
Dialog appropriateness	M=4.89, SD=0.72	M=3.88, SD=1.40	0.006
Learning ability	M=5.79, SD=0.80	M=3.21, SD=1.73	4.79×10^{-7}
Usefulness	M=5.16, SD=1.24	M=3.03, SD=1.80	9.36×10^{-5}
Long-term potential	M=4.98, SD=1.29	M=3.74, SD=1.66	0.012

Table 5.9: Pearson's r for correlations of aspects of user experience and the subject impression for memory

Measures	r	p
Learning ability	0.89	4×10^{-6}
Usefulness	0.81	0.0001
Long-term potential	0.73	0.0001
Dialog appropriateness	0.71	0.002
User engagement	0.59	0.017
User responsiveness	0.26	0.338

Discussions

The first hypothesis is fully accepted. Both the objective measure and the subjective measure indicate that the proposed MCAEM model has improved the retrieval accuracy given free dialog inputs. This is because our memory model can flexibly retrieve previous experience. In free dialog interactions, users usually expressed the same semantic meaning by different words or the different meanings by the same words. The consideration of the relations between elements in the similarity measure works compatibly with the free dialog scenario. Although Sophie without our memory component still had certain learning ability from the chatbot, the strict requirement for the sentence formats drastically degraded its performance in free communication. The subjective impression shows the same tendency as the objective measure. It indicates that episodic memory models with better retrieval accuracy can improve user impressions for the memory capabilities. In addition, the participants feel less distinctiveness of the retrieval accuracy than the real case. This may be because

users tend to have default expectation that the memory capability should come along with the human figure. Therefore, the memory capability of Sophie without the memory component has been over-estimated.

The second hypothesis is partially accepted. The improvement in *dialog appropriateness* is due to the close relation of the retrieved previous utterances to the current context. This enhances the logic coherence of the dialogs. However, the occasional speech recognition errors and unclear utterances of the text-to-speech component are reported that largely confuse the participants. This to some extent decreases the difference in *dialog appropriateness* of the two versions. The significantly strengthened *learning ability* and *usefulness* are results of the enhanced ability to remember previous discourses and to make associations of peer users. Some participants reported that they found it amazing that Sophie showed much knowledge of their friends in the conversations. Users have more willings to have further interaction with Sophie with the memory component because they find more interesting to interact with Sophie that can grow with interactions and have increasing understanding of them. The significant differences in *user engagement* and *user responsiveness* are not observed may because they have less dependence on memory performance. Other factors such as an appealing appearance, believable facial expressions or gestures and attractive voices of the social companion may play important roles too.

Finally, the third hypothesis is fully accepted. It takes some explanation role for the second Hypothesis. The results indicate that the observed memory performance to some extent has positive influence on user experience.

Chapter 6

BMS: A Benchmark from Movie Scripts for Episodic Memory Retrieval

In this chapter, we propose a Benchmark from Movie Scripts (BMS) evaluating retrieval performance of episodic memory models designed for free-dialog based human-companion interactions.

6.1 Introduction

We target at building a Benchmark from Movie Scripts (BMS) for episodic memory retrieval. The benchmark is designed for episodic memory models in the context of free-dialog based companion-user interactions. Prior studies of episodic memory in this context have rarely presented experimental comparisons with other models. Many of them only compare with related work conceptually and focus on the justification of their own models. This is partly caused by the lack of available benchmarks for this kind of episodic memory models.

There exists some benchmarks which evaluate episodic memory models using simple domains in artificial intelligence applications. The Monroe Plan Corpus [10], which was originally designed for plan recognition, has been used to evaluate episodic memory mod-

els [121, 66]. This dataset contains 5000 episodes generated by a HTN (Hierarchical Task Network) SHOP2 (Simple Hierarchical Ordered Planner 2) Planner in a predefined logistic domain. Each episode is constituted by sequential subtasks and methods for random goals and starting states. The Unreal Tournament 2004 (UT2004) dataset has been adopted for the evaluation of the EM-ART model [135]. The episodes of this dataset are collected from the UT2004 first-person shooting game. Each episode contains all events in one battle. In addition, Rudolf et.al [67] build two datasets for episodic memory: Behan-HTN and Cermak-3D. The Behan-HTN dataset contains episodes that are generated in the context of human activities by adapting the HTN planner of the Monroe Plan Corpus [10]. The Cermak-3D dataset is a collection of agent activities generated by an AND-OR tree planner in 3D virtual environment. However, all these benchmarks are generated in simple and small domains. These domains with limited number of goals, tasks, states, methods and rewards are too simplified to model real-world episodic memory problems.

The domain of free-dialog based companion interactions is much more complicated. The scale of attributes in this domain is quite large due to various topics in free dialogs and tremendous words from natural language. Moreover, the inputs from natural language may have noises from speech or recognition errors, flexible sentence structures and big amount of synonymies and polysemies.

In this chapter, we build the BMS benchmark to measure and compare retrieval performance of episodic memory models for free-dialog based companion-user interactions. The BMS benchmark dataset contains 38,058 attributes, 76,123 events and 4,889 episodes. The large-scale of attributes makes the benchmark complicated. Episodes of the BMS benchmark are extracted from 106 movie scripts which cover various topics in daily life. Three models, the MCAEM model in Chapter 6, ISAC model [34], and EM-ART model [136], are evaluated by their retrieval accuracies and efficiencies for events and episodes. To test error resistances of these models, the BMS benchmark provides four types of retrieval cues: complete cues, noisy cues, partial cues and noisy partial cues. We analyze the consistency between results of executing different models on the BMS benchmark and on previous

Figure 6.1: An example part of movie script. The attributes to be extracted are denoted in red boxes and words.

benchmarks. In addition, we also compare the proposed MCAEM model with the other two models on our BMS benchmark.

6.2 Acquisition of Dataset

It is challenging to build an episodic memory benchmark for free-dialog based companion-user interactions, because collecting large-scale episodes from real-world interactions is tedious and time-consuming. To collect these episodes manually, one needs not only a plenty of participants but abundant well-designed topics and contexts of interactions. Instead of collecting episodes from real-world interactions, we construct the BMS benchmark by extracting episodes from hundreds of movie scripts. The movie scripts are selected to replace real-world interactions for the following reasons: (1) The episodes collected from movies

are complicated enough to simulate real-word interactions. Movie scripts contain natural dialog interactions for various contexts and topics in daily life. The episodes from movies have complex story lines and turn takings, and have sentences with ambiguous semantic meanings that need to be understood with contexts. (2) The scale of attributes, events and episodes that can be collected from movie scripts is large enough. (3) Movie scripts are usually written in certain formats that makes the design of extraction rules easy.

The BMS benchmark dataset is constructed based on 106 movie scripts crawled from online websites*. The collected movie scripts cover widespread topics about school, office, prison, adventure, love, etc. We design extraction rules for various formats of movie scripts. The extracted episodes are constituted by events that happened in the same contexts. For each event, we extract actor's lines and context attributes such as subject, objects, times, locations and emotion polarities. The emotional polarities of events are computed by a sentiment analysis tool[§]. An example part of a movie script is shown in Figure 6.1, and an example episode extracted from movie scripts is displayed in Figure 6.2. It is obvious that the talking scenario in this episode has high probability to happen in real-world interactions.

After filtering those episodes having less than three events, we get 4,889 episodes in total in the BMS benchmark, which contain 38,058 attribute values and 76,123 events. The BMS benchmark has much more complicated episodes compared to those collected from the small domains discussed before, because tremendous words in natural language have to be considered as attributes.

6.3 Platform for Evaluation

Episodic memory models are supposed to retrieve relevant events or episodes given appropriate cues. Our BMS benchmark provides experiments to test retrieval performance of models for events and episodes. The retrieval performance contains two aspects: the

*dailyscript.com

[§]<https://pypi.python.org/pypi/senticnet>

1	Subject	Object	Location	Time	Polarity	Sentence
2	KAT	BIANCA	GIRLS' ROOM	DAY	0.03	I said, leave it!
3	BIANCA	KAT	GIRLS' ROOM	DAY	0.06	Did you change your hair?
4	CHASTITY	BIANCA	GIRLS' ROOM	DAY	0.0	No.
5	BIANCA	CHASTITY	GIRLS' ROOM	DAY	0.06	You might wanna think about it
6	BOY	BIANCA	GIRLS' ROOM	DAY	0.0	Hey, Bianca.
7	GIRL	CAMERON	GIRLS' ROOM	DAY	0.82	Awesome shoes.
8	MISS PERKY	GIRL	GIRLS' ROOM	DAY	0.95	I'm sure you won't find Padua any different
9	CAMERON	MISS PERKY	GIRLS' ROOM	DAY	-0.06	I don't think so, ma'am
10	MISS PERKY	PATRICK	GIRLS' ROOM	DAY	0.3	Then go forth. Scoot I've got deviants
11	PATRICK	MISS PERKY	GIRLS' ROOM	DAY	-0.89	I missed you.
12	MISS PERKY	PATRICK	GIRLS' ROOM	DAY	0.06	It says here you exposed yourself to a gr
13	PATRICK	MISS PERKY	GIRLS' ROOM	DAY	0.0	It was a bratwurst. I was eating lunch.
14	MISS PERKY	PATRICK	GIRLS' ROOM	DAY	-0.03	With the teeth of your zipper?
15	MICHAEL	MISS PERKY	GIRLS' ROOM	DAY	0.14	You the new guy?
16	CAMERON	MICHAEL	GIRLS' ROOM	DAY	0.0	So they tell me...
17	MICHAEL	CAMERON	GIRLS' ROOM	DAY	0.0	C'mon. I'm supposed to give you the tour. s
18	CAMERON	MICHAEL	GIRLS' ROOM	DAY	0.14	North, actually. How'd you ?
19	MICHAEL	CAMERON	GIRLS' ROOM	DAY	0.39	I was kidding. People actually live there?
20	CAMERON	MICHAEL	GIRLS' ROOM	DAY	0.09	Yeah. A couple. We're outnumbered by the c
21	MICHAEL	CAMERON	GIRLS' ROOM	DAY	0.08	How many people were in your old school?
22	CAMERON	MICHAEL	GIRLS' ROOM	DAY	0.0	Thirty-two.
23	MICHAEL	CAMERON	GIRLS' ROOM	DAY	0.0	Get out!
24	CAMERON	MICHAEL	GIRLS' ROOM	DAY	0.1	How many people go here?
25						

Figure 6.2: An example episode extracted from movie scripts

average retrieval accuracy and average retrieval time.

In our BMS benchmark, experiments are conducted given several types of retrieval cues. Each experiment tests candidate models in 20 trials. In each trial, we calculate the average evaluation result using 100 randomly generated cues. The workflow of the evaluation platform is shown in Figure 6.3. In following sections, we will introduce the automatic generation of different types of retrieval cues and the selection of evaluation metric.

6.3.1 Automatic Generation of Retrieval Cues

To test retrieval performance, we generate retrieval cues from randomly selected events or episodes from the BMS benchmark dataset. These source events or episodes simultaneously become the targets of retrievals. In the BMS benchmark, four types of cues are generated: (1) complete cue, (2) partial cue, (3) noisy cue, (4) noisy partial cue.

Complete cues are exact duplications of the source events or episodes. The complete cues are mainly not used to test models but to check if models are correctly setup, because episodic memory models usually have perfect retrieval performance given complete cues.

Instead of inviting experts to manually determine erroneous cues (including the noisy

Figure 6.3: The workflow for the evaluation platform

cues, partial cues and noisy partial cues) and ground truths, we automatically generate erroneous cues and take the source events or episodes as ground truths. Nevertheless, naive generation of erroneous cues could be problematic if noises, especially big ones, dramatically change the original semantic meanings of the source events or episodes so that they should not be the ground truths anymore (however they are still taken as the ground truths in performance measure). A reasonable solution is to preserve the important attributes of the source events or episodes that keep the main semantic meanings unchanged. The important attributes can be identified by the TF-IDF indexing [110]. Our erroneous cue generations below take the importances of attributes into consideration.

Partial cues are parts of the source events or episodes. The total numbers of attributes or events in these partial cues are controlled by those of the source events or episodes multiplied by a ratio specified by users. Partial cues for events can be generated by selecting attributes from the source events following the order of importances (of attributes). Likewise, partial cues for episodes can be generated by randomly selecting events from source episodes. Events can either be selected in continuous or noncontinuous order, while the latter should still keep the same relative order of events (in the source episodes).

To simulate noisy inputs from users in read-world free-dialog based interactions, noisy cues are generated by substituting word attributes and context attributes in source events or

episodes with the alternative ones with similar semantic meanings. The real-world noises come from both speech recognition errors and large amount of synonymies (multiple words share the same meaning) and polysemies (one word has multiple meanings) in natural language. We measure similarities of word attributes and context attributes by training two latent semantic analysis (LSA) models. The two LSA models are trained based on documents of the latest Wikipedia corpus and episodes in our BMS benchmark dataset respectively. In noisy cue generation, the substitution of attributes follows two disciplines: (1) attributes are considered for substitution if and only if their similarities with the closest alternatives are beyond a threshold specified by users. (2) Key attributes have a stricter threshold for substitution.

Noisy partial cues are generated by first select parts of source events or episodes followed by giving noises. The generation follows the same disciplines discussed above.

6.3.2 Evaluation Metric

To evaluate retrieval performance on the BMS benchmark dataset, we adopt the mean reciprocal rank (MRR) [81] measurement. MRR computes average of n retrieval attempt scores in the range of 0 and 1. The scores are reciprocals of ranks ($rank_i$) of target objects in the list of retrieved objects. As discussed above, the targets are the source events or episodes used for cue generation. The computation of MRR is shown below:

$$MRR = \frac{1}{n} \sum_{i=1}^n \frac{1}{rank_i} \quad (6.1)$$

The ranks of retrieved events or episodes need to be specially treated when duplications happen. The results are the lowest ranks of the duplicated events or episodes with the targets. Higher MRR values indicate better retrieval accuracies of episodic memory models.

6.4 Models for Comparison

Three episodic memory models are selected to make comparison on the BMS benchmark: (1) ISAC model [34], (2) EM-ART model [136], and (3) our MCAEM model. We select the ISAC model and EM-ART model for comparison because the algorithms in this model are clearly described for implementation. Although these models have been briefly introduced in section 2.3.3, here we present more details of these models and their parameter settings in the experiments below.

6.4.1 ISAC Model

ISAC model takes episodes as collections of Semantic Memory (SM) units (namely, the term “attribute values” we used in this thesis). The ISAC model contains two main processes: training and retrieval.

Training

The training of ISAC is to index episodes by their “fingerprint”. Fingerprints are defined as lists of normalized weighted occurrence frequencies of SM units in episodes. As displayed in Figure 6.4 [33], the computation of the fingerprint has several steps below:

- (1) **Collect occurrence frequency** For episodes in dataset, collect statistics of occurrence frequencies of SM units in episodes. The frequency of a SM unit s in an episode ep is denoted as c_s .
- (2) **Calculates weights for SM units** We compute the weight w_s of the SM unit s by $w_s = B/n_s$. n_s is the total occurrence of s in all episodes. B is the intense of attention given to s .
- (3) **Calculate “fingerprint” for each episode** The “fingerprint” of an episode is constructed to be a hash table of all SM units in the episode corresponding to their fingerprint val-

Figure 6.4: The computation of fingerprint in ISAC model

ues. The fingerprint value F_s^{ep} of the SM unit s in the episode ep is calculated by $F_s^{ep} = w_s c_s / N_{ep}$, where N_{ep} is the total number of SM units in ep .

Episode Retrieval

The ISAC can retrieve relevant episodes given cues. The algorithm of episode retrieval is described below:

1. Compute fingerprint L for a given retrieval cue c . Sort SM units in L by their fingerprint values in decreasing order. Denote N as the total number of episodes to be retrieved and R as an empty hash to store relevant scores of candidate episodes.
2. Get all episodes that contain the SM unit L_i , where i starts from 0. Calculate relevant score R_{ep} for each episode ep . We have $R_{ep} = F_{L_i}^{ep} F_{L_0}^c H_{ep}$. $F_{L_i}^{ep}$ and $F_{L_i}^c$ are the fingerprint values of L_i in the episode ep and the cue c , H_{ep} is the historical value of

ep . The historical value is related to the total recall of ep and the emotional salience of ep , and it decays with time exponentially.

3. Update R by episodes with their relevant scores R_{ep} . Sort episodes in R in decreasing order
4. If R contains more than N episodes. Filter out extra episodes ep (starting from the index $N + 1$) on the condition that $R_N - R_{ep} > L_{i+1}$. After that, if R is still longer than N , continue the filtering process if $R_N - R_{ep} > L_{i+1}/H_{ep}$.
5. If R contains only N episodes or i reaches the end of L , return all episodes in R . Otherwise repeat the step 2 to step 5 by setting $i = i + 1$.

In this implementation, we set parameters of the ISAC model as follows: (1) the intense of attention B for all SM units is set to 1, (2) The total number of episodes to be retrieved N is set to 20. (3) In the experiments of our BMS benchmark, the random cue generation induces no bias in the frequency and recency of episodes. Thus we don't consider the total recall and the time effect in the historical value H_{ep} . The simplified version of historical value is $H_{ep} = I_{ep}/3$, where I_{ep} is the emotion salience of ep computed by the normalized emotion polarities of events in ep .

6.4.2 EM-ART Model

The mechanism of the EM-ART model has been introduced in the section 5.2. In this implementation, the attribute layer F_1 has two input fields: context attribute values and words of character lines. Table 6.1 displays the parameter settings for the event and episode channels. The parameters are the same for all input fields in each layer. The vigilances are set to 1s to follow the optimized settings claimed by the authors of the EM-ART model.

Table 6.1: The parameters of EM-ART model on the proposed BMS benchmark

Channel	α^k	β^k	γ^k	ρ^k
Event Channel	0.1	0.3	1	1
Episode Channel	0.1	0.3	1	1

6.4.3 MCAEM Model

The setting of our MCAEM is presented as follows: Firstly, attributes of events are separated into word attributes and context attributes. The numbers of clusters for word attributes and context attributes are set to 1,000 and 500 respectively, and finally we get 4,106 clusters for events. In addition, the number of episode clusters is set to 500.

6.5 Experiments

In this section, we analyze the consistency between the results of testing different models on our BMS benchmark and on the Monroe [10] and UT2004 dataset (shown in section 5.4.2). In addition, we also compare the proposed MCAEM model with the EM-ART model [136] and ISAC model [34] by evaluating them on our BMS benchmark.

6.5.1 Experiment Setup

We conduct experiments on two retrieval tasks: event retrieval and episode retrieval. In event retrieval, we only consider the EM-ART model and our MCAEM model because the ISAC model [34] only provides algorithm for episode retrieval. As mentioned in Section 6.3.1, the models are tested by noisy cues, partial cues and noisy partial cues. In the first experiment to evaluate consistency of our benchmark, the ratio of partial cues is set to 70% and the threshold of noisy cues is set to 0.7. While in the second experiment to compare models on our BMS benchmark, the parameters are set in multiple levels. Each experiment tests candidate models given types of cues for 20 trials. In each trial, we calculate the MRR values given 100 randomly generated cues.

Table 6.2: Comparing the scales of the tested datasets

Channel	Monroe [10]	UT2004	BMS
#(Attributes)	106	240	38,058
#(Events)	50,223	17,579	76,123
#(Episodes)	5000	394	4889

6.5.2 Results

Comparison with Other Datasets

We compare results of the proposed BMS benchmark versus those achieved with the Monroe and UT2004 dataset, which have been used to validate prior episodic memory models [121, 66, 135].

The complexities of episode datasets are highly related to the number of attributes, which determines the upper bounds of the possible distinct events and episodes in the datasets. According to the scales of the three datasets displayed in Table 6.2, the Monroe and UT2004 dataset are much more homogeneous and redundant due to their limited numbers of attributes. Many events and episodes in these two datasets are duplicated or very close to each other.

We consider the benchmark consistency by the relative retrieval performance of models against the average on each dataset. The BMS benchmark is consistent to the prior ones if the relative performance of models follow the same tendency. The relative retrieval performance is measured by the normalized MRR (nMRR) which scales the MRR values down by the average over all models tested and shifts the average to zero:

$$nMRR_m = MRR_m / AvgMRR - 1 \quad (6.2)$$

where MRR_m is the MRR value computed for the model m , and AvgMRR is the average MRR value for all models in the same experiment.

The normalized MRR scores are computed for the three models on the three different

Table 6.3: Evaluating the event retrieval of models on the three benchmarks. The normalized MRR (nMRR) measure is adopted to compare the relative performance of models on the three benchmarks

Model	Noisy Cue			Partial Cue		
	Monroe [10]	UT2004	BMS	Monroe [10]	UT2004	BMS
MCAEM	-3.8%	2.9%	19.1%	-29.0%	-7.2%	1.1%
EM-ART	3.8%	-2.9%	-19.1%	29.0%	7.2%	-1.1%

Table 6.4: Evaluating the episode retrieval of models using different datasets. The normalized MRR (nMRR) measure is adopted to compare the relative performance of models on the three benchmarks

Model	Noisy Cue			Partial Cue		
	Monroe [10]	UT2004	BMS	Monroe [10]	UT2004	BMS
MCAEM	54.5%	47.7%	29.6%	31.6%	22.1%	2.7%
EM-ART	28.0%	19.6%	12.1%	25.2%	12.0%	2.5%
ISAC	-82.5%	-67.2%	-41.7%	-56.7%	-34.1%	-0.5%

datasets. Table 6.3 and Table 6.4 show the results of event and episode retrievals given noisy and partial cues.

Comparison of Models on the BMS Benchmark

As the supplemental evaluations for the proposed MCAEM model (in the Chapter 5) in the context of free-dialog based social companion interactions, we also analyze and compare the results of the three models on the BMS benchmark dataset. The parameters for erroneous cue generation are set to multiple levels. The ratio of partial cues is set to 70%, 50% and 30%, and the threshold of noisy cues is set to 0.9, 0.7 and 0.5. The smaller partial ratio and noisy threshold result in cues with higher erroneousness. The results of the experiments in event and episode retrievals are deliberated below.

Event Retrieval This experiment compares the EM-ART model and our MCAEM model in event retrieval. Figure 6.5 illustrates the average retrieval accuracy of the two models with multiple levels of erroneousness. The MCAEM model has better error resistance than the EM-ART model given noisy and partial cues at different levels. Both models have dropped performance with the increase of erroneousness, however, the performance of the

Figure 6.5: Results of event retrieval given multiple levels of erroneousness. The ratio of partial cues is set to 70%, 50% and 30%, and the threshold of noisy cues is set to 0.9, 0.7 and 0.5. The blue and red colors stand for the MCAEM and EM-ART model respectively.

EM-ART model decreases faster. More specifically, in the specific setting of partial rate as 70% and noise threshold as 0.7, Figure 6.6 displays the average retrieval accuracy in the 20 trials on conditions of noisy cues (Figure 6.6(a)), partial cues (Figure 6.6(b)), noisy partial cues (Figure 6.6(c)) and average retrieval time (Figure 6.6(d)). Two-tail T-test analyses are conducted. The results indicate that the two models have significant different retrieval accuracy given noisy cues ($t = 19.127, p < \times 10^{-16}$), partial cues ($t = 2.884, p = 0.0066$) and noisy partial cues ($t = 27.374, p < 2.2 \times 10^{-16}$), and have significant different retrieval time ($t = -68.866, p < 2.2 \times 10^{-16}$). In overall, the MCAEM model has better retrieval accuracy and shorter retrieval time than the EM-ART model.

Episode Retrieval This experiment makes comparison of our MCAEM model, ISAC model and EM-ART model in episode retrieval. Figure 6.7 illustrates the average retrieval accuracy of episode retrieval given multiple levels of noise and incompleteness of cues. The three models have robust performance given partial cues at different levels. Meanwhile, the MCAEM shows better results and less performance decrease than the other two models in all levels of noisy cues. We conduct statistical analyses for the results of episode retrieval given noisy cues, partial cues and noisy partial cues with partial rate as 70% and noise

(a) Noisy cues

(b) Partial cues

(c) Noisy partial cues

(d) Retrieval Time

Figure 6.6: Results of event retrieval given various cues when the ratio of partial cues is set to 70% and the threshold of noisy cues is set to 0.7. The blue and red colors stand for the MCAEM and EM-ART model respectively.

threshold as 0.7. The ANOVA analysis demonstrates that the retrieval accuracy of the three models have significant difference for noisy cues ($F = 1572.04, p < 2.2 \times 10^{-16}$) display in Figure 6.8 (a), partial cues ($F = 239.39, p < 2.2 \times 10^{-16}$) in Figure 6.8 (b), noisy partial cues ($t = 1457.89, p < 2.2 \times 10^{-16}$) in Figure 6.8 (c). In addition, the three models also have significant different average retrieval time ($F = 29.68, p = 2.22 \times 10^{-16}$) in Figure 6.8 (d). In overall, the retrieval accuracies of the three models are ranked as follows: the MCAEM model is better than the EM-ART model, which is better than the ISAC model. One exception case is that the EM-ART model and the MCAEM model have no significant different results for the partial cues by two-tail T-test ($t = 1.78, p = 0.089$). In addition, the ISAC model has used much less average retrieval time than the other two models.

Figure 6.7: Results of episode retrieval given multiple levels of erroneousness. The ratio of partial cues is set to 70%, 50% and 30%, and the threshold of noisy cues is set to 0.9, 0.7 and 0.5. The blue, red and green colors stand for the MCAEM, EM-ART and ISAC model respectively.

6.5.3 Discussion

Comparison with Other Datasets As shown in Table 6.4, the BMS benchmark is consistent with the two other datasets in episode retrievals for noise and partial cues. Compared to the two other datasets, models on our BMS benchmark have no big difference in results for partial cues, because our BMS benchmark contains rare episodes that are extremely close to each other. On the contrary, homogeneous datasets built on simple domains, such as the Monroe and UT2004, can better evaluate the event-order robustnesses of models due to the large amount of similar event sequences in episodes.

However, homogeneous datasets have problem in evaluating event retrievals. The shortness of events compared to episodes makes them relatively easier to be significantly changed by noises or incompleteness. Because of the high density of similar events in homogeneous datasets, source events are probably turned into another given even slight noises or incompleteness. Thus it could be problematic to continue taking source events as the retrieval targets for noisy and partial cues. This explains the inconsistency of event retrievals on our BMS benchmark with the two other (homogeneous) datasets as shown in Table 6.3.

Based on above analysis, our BMS benchmark is consistent with existing datasets for

Figure 6.8: Results of episode retrieval given various cues when the ratio of partial cues is set to 70% and the threshold of noisy cues is set to 0.7. The blue, red and green colors stand for the MCAEM, EM-ART and ISAC model respectively.

episode retrievals and is more suitable to test event retrievals compared to homogeneous datasets.

Comparison of Models on the BMS Benchmark While the comparisons of models are not the main concern of the BMS benchmark, the results provide interesting additional evaluation for the MCAEM model proposed in Chapter 5. The results of event and episode retrievals indicate that the proposed MCAEM model has more accuracy in retrieval than the EM-ART model and ISAC model given cues with noises and incompleteness. Although the proposed event-order measure of the MCAEM model is not significant better than the EM-ART model on our BMS benchmark dataset due to the lack of extremely similar episodes,

it shows significant better performance on the Monroe and UT2004 dataset. The better performance of our MCAEM model results from its comprehensive similarity measure. The similarity measure of the MCAEM model improves error resistance in several aspects: (1) it gives more weights for key attributes to preserve the main semantic meanings, (2) it tolerates the replacement of attributes with similar alternatives (synonymies for words) by considering relations of attributes, (3) it provides more robust comparison for the event-order in episodes. In addition, the ISAC model has worse performance in episode retrievals because it never considers the order of events. The retrieval accuracy of the EM-ART model has dropped a lot for noisy partial cues, because the EM-ART model with vigilance as 1 is close to the boolean retrievals (with event-order measure) on our BMS benchmark, the performance of which heavily relies on the amount of shared elements of cues and targets.

Chapter 7

Conclusions and Future Work

In this thesis, we have developed a general system architecture for social companions which is shared by our virtual human Sophie and social robot Nadine. Under the system architecture, we focused on the affective system and episodic memory module. We have proposed novel algorithms for the two modules which are validated by comprehensive experiments and user studies. In Section 7.1, we present our contributions with respect to our goals as mentioned in Chapter 1. Despite the promising results that have been achieved, there are still some limitations in our work. Section 7.2 discusses these limitations and possible directions for future work.

7.1 Contributions

In this section, we present the significance of our work, contributions to the state of the art, and lessons we have learned.

Design system architecture for social companions We have presented a system architecture for social companions, shared by our virtual human Sophie and social robot Nadine. The perception, internal processing and action are the three main components. The architecture is designed to achieve easy plugging-in of modules and low-cost communications of

the modules. The communication and synchronization of these modules are carefully designed, especially for the affective system and episodic memory which are critical for personalization of social companions, to integrate their functionalities to achieve higher-level capabilities. This architecture can benefit future researchers to build their own personalized social companions from scratch.

Design reasonable objective function for mood dynamics We have presented a personality-characterized mood dynamics (PCMD) model for social companions. Prior studies model mood dynamics without clear objectives. Our contribution is that we proposed a clear and reasonable objective function for mood dynamics model to strengthen the relations of moods and personalities. Our model specifies how particular personalities influence interactions between emotions and moods so that the overall mood will approach personalities after sufficient interactions. The conducted numerical simulations, case studies and user studies demonstrate the convergence of the PCMD model and its effectiveness in personalizing social companions. More specifically, given well-designed emotional behaviors and rules for affect-driven behavior selection, the PCMD model can combine the emotional behaviors in certain patterns to show particular personalities of social companions. This work inspires us that future researches can specify different objective functions of mood dynamics to achieve particular effects in companion interactions.

Propose mixed correlation analysis for episodic memory We have presented a mixed-correlation-analysis based episodic memory (MCAEM) to improve accuracy of retrieval. We make analysis of correlations of memory elements in three aspects: the relations between elements, the weights of attributes, and the order of events. The proposed mixed correlation analysis have following novel considerations: (1) the weights of attributes are introduced based on not only their occurrence frequencies but also their specialness in episodes; (2) besides considering the relations across element levels, the relations of the same element levels are also considered. (3) more balanced and robust analysis is per-

formed for event-order correlations in episodes. Based on the mixed correlation analysis, a new similarity measure is proposed, which has substantially improved the performance of the model. The experiments in the UT2004 platform and the user study for companion interactions indicate the effectiveness and generality of our model. Our model can both improve agent performance at particular tasks and can enhance user experience in companion interactions.

Build a benchmark for episodic memory retrieval in free-dialog based companion interactions We have built a benchmark from movie scripts (BMS) for episodic memory models in the context of free-dialog based companion interactions. The episodes in the benchmark are collected by extracting attributes and events from 106 movie scripts crawled online. The database contains 38,058 attribute values, 76,123 events and 4,889 episodes, which are more realistic and complicated than those from homogeneous databases built on small domains. The benchmark also provides evaluation metric and automatic generation of retrieval cues with noise and incompleteness to simulate real-world interactions. The conducted experiments demonstrate that our benchmark is consistent with existing databases for episode retrievals and is more suitable to test event retrievals compared to homogeneous databases. The results also indicate that our MCAEM model has significant better error resistance compared to two previous models on this benchmark. This work provides a benchmark for the research community to compare and improve episodic memory models for free-dialog based companion interaction.

7.2 Future Work

In this section, we discuss limitations in our work and present several directions to be further improved in future work.

Possible improvements and applications of PCMD model For the personality-characterized mood dynamics (PCMD) model proposed in Chapter 4, it has several limitations. Firstly, the convergence of the numerical approach for the PCMD model is slow, an acceleration for the algorithm is demanded. Secondly, the deficiency of emotional behaviors and the lack of types of emotions generated by appraisal have great limitation on the believability of social companion interactions. Thus, we will design more expressive emotional behaviors and include more types of emotions for our social companions. Thirdly, the probability distribution of emotions $\{p_i\}_{i=1}^S$ is fixed in current applications. One future work is to adjust behavior patterns by computing emotion weights periodically based on the dynamics of emotion distribution changing with the progress of interactions. Finally, our future research will also consider mutual interactions of multiple social companions with different personalities. It is interesting to explore how an easy-going virtual human and a tough social robot interact with each other and their different emotional reactions to the same events

Explore optimized settings for MCAEM model The proposed mixed correlation analysis in Chapter 5 considers correlations of memory elements in three aspects: the relations between elements, the weights of attributes and the order of events. In the thesis, we just prove that combination of these aspects in similarity measure has achieved good effects in retrieval accuracy. In future work, we will make analysis of each aspect individually to identify which aspect takes main effects in what situation. The individual analysis can help us understand the mechanism better.

In addition, the pipeline of building the hierarchical relational space contains multiple components including the discretization, the embedding, the clustering. In the thesis, we select the fuzzy discretization, the LSA for embedding and the Kmeans for clustering. However, many other possibilities should be explored in future work. For example, how about using the crisp discretization which separates continuous domains into several crisp sets with binary memberships. What about using the Word2Vec method [90] for embedding and using methods such as the K-modoids [103] or the affinity propagation (AP) [42] for

clustering. Moreover, many parameter settings also need to be optimized. For example, we will explore optimized number of clusters for Kmeans and optimized number of retrieved clusters for event and episode retrievals. Experiments will be conducted to test the impacts of each component and parameter for retrieval accuracy given various cues.

Extend episodic memory for multi-party interactions To improve believability and naturalness of social companions, our research will put more focus on memory functionalities for multi-party interactions. The new functionalities to explore include information sharing, memory filtering and privacy protection.

As parts of whole system, it is easy but not natural to enable all companions to have access to everything. Each companion should keep its own memories by filtering information. For example, if a virtual human never met a robot before, they should interact like strangers instead of knowing everything about each other. In addition, social companions will have different memory records even for the same events due to different memory filtering and viewpoints. They need to share their knowledge and information to each other in cooperation if necessary. Another important aspect in social interaction is privacy protection, whereby social companions should protect privacy information for themselves and their friends. However, it is difficult to achieve this functionality. For example, how to judge which memory should be kept and which one should be filtered; how to judge which memories can be shared to whom and which cannot; how to share information in a more efficient way. All these questions remain open nowadays.

In future research, in addition to episodic memory, we plan to study computational models for procedural memory that is best adapted as representing acquired skills, and semantic memory that enables virtual humans or social robots to memorized knowledge and concepts based on facts.

References

- [1] C. Adam, L. Cavedon, and L. Padgham. "Hello Emily, how are you today?": personalised dialogue in a toy to engage children. In *Proceedings of the Workshop on Companionable Dialogue Systems*, pages 19–24. Association for Computational Linguistics, 2010.
- [2] J. P. Aggleton and M. W. Brown. Interleaving brain systems for episodic and recognition memory. *Trends in cognitive sciences*, 10(10):455–463, 2006.
- [3] S. A. Ahadi and M. K. Rothbart. Temperament, development, and the big five. *The Developing Structure of Temperament and Personality from Infancy to Adulthood*, pages 189–207, 1994.
- [4] G. W. Allport. *Pattern and Growth in Personality*. New York: Holt, Rinehart & Winston, 1961.
- [5] C. O. Alm, D. Roth, and R. Sproat. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 579–586. Association for Computational Linguistics, 2005.
- [6] E. B. Andrade. Behavioral consequences of affect: Combining evaluative and regulatory mechanisms. *Journal of Consumer Research*, 32(3):355–362, 2005.

-
- [7] A. Beck, Z. Zhijun, and N. Magnenat-Thalmann. Motion control for social behaviors. In *Context Aware Human-Robot and Human-Agent Interaction*, pages 237–256. Springer, 2016.
- [8] C. Becker, S. Kopp, and I. Wachsmuth. Simulating the emotion dynamics of a multi-modal conversational agent. In *Affective Dialogue Systems*, pages 154–165. Springer, 2004.
- [9] C. Becker-Asano and I. Wachsmuth. Affect simulation with primary and secondary emotions. In *Intelligent Virtual Agents*, pages 15–28. Springer, 2008.
- [10] N. Blaylock and J. Allen. Generating artificial corpora for plan recognition. In *User Modeling*, pages 179–188. Springer, 2005.
- [11] G. H. Bower. Mood and memory. *American Psychologist*, 36(2):129, 1981.
- [12] S. Brave and C. Nass. Emotion in human-computer interaction. In *The Human-Computer Interaction Handbook*, pages 81–96. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 2003.
- [13] C. Brom, T. Korenko, and J. Lukavský. How do place and objects combine?”what-where” memory for human-like agents. In *Intelligent Virtual Agents*, pages 42–48. Springer, 2009.
- [14] C. Brom, K. Pešková, and J. Lukavský. What does your actor remember? towards characters with a full episodic memory. In *Virtual Storytelling. Using Virtual Reality Technologies for Storytelling*, pages 89–101. Springer, 2007.
- [15] J. M. Burger. *Personality*. Belmont: Thomson Wadsworth, 2008.
- [16] R. Calvo, S. D’Mello, et al. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing*, 1(1):18–37, 2010.

-
- [17] E. Cambria and A. Hussain. *Sentic computing*. Springer, 2012.
- [18] J. Campos and A. Paiva. May: My memories are yours. In *Intelligent Virtual Agents*, pages 406–412. Springer, 2010.
- [19] L. Cañamero. Emotion understanding from the perspective of autonomous robots research. *Neural networks*, 18(4):445–455, 2005.
- [20] J. Cassell, T. Bickmore, L. Campbell, K. Chang, H. Vilhjálmsón, and H. Yan. Requirements for an architecture for embodied conversational characters. In *Computer Animation and Simulation99*, pages 109–120. Springer, 1999.
- [21] G. Castellano, L. Kessous, and G. Caridakis. Emotion recognition through multiple modalities: face, body gesture, speech. In *Affect and emotion in human-computer interaction*, pages 92–103. Springer, 2008.
- [22] G. L. Clore and K. Gasper. Feeling is believing: Some affective influences on belief. *Emotions and Beliefs: How Feelings Influence Thoughts*, pages 10–44, 2000.
- [23] M. Coulson. Attributing emotion to static body postures: Recognition accuracy, confusions, and viewpoint dependence. *Journal of nonverbal behavior*, 28(2):117–139, 2004.
- [24] S. Danilava, S. Busemann, and C. Schommer. Artificial conversational companions a requirements analysis. In *International Conference on Agents and Artificial Intelligence*, pages 282–289, 2012.
- [25] T. Danisman and A. Alpkocak. Feeler: Emotion classification of text using vector space model. In *Convention Communication, Interaction and Social Intelligence*, volume 2, pages 53–59, 2008.
- [26] R. J. Davidson. On emotion, mood, and related affective constructs. *The Nature of Emotion: Fundamental Questions*, pages 51–55, 1994.

-
- [27] F. De la Torre, W.-S. Chu, X. Xiong, F. Vicente, X. Ding, and J. F. Cohn. Intraface. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 2015.
- [28] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.
- [29] L. Devillers and L. Vidrascu. Real-life emotions detection with lexical and paralinguistic cues on human-human call center dialogs. In *Interspeech*, 2006.
- [30] J. Dias, W. C. Ho, T. Vogt, N. Beeckman, A. Paiva, and E. André. I know what I did last summer: Autobiographic memory in synthetic characters. In *Affective Computing and Intelligent Interaction*, pages 606–617. Springer, 2007.
- [31] J. Dias and A. Paiva. Feeling and reasoning: A computational model for emotional characters. In *Portuguese Conference on Artificial Intelligence*, pages 127–140. Springer, 2005.
- [32] S. D’Mello and A. Graesser. Automatic detection of learner’s affect from gross body language. *Applied Artificial Intelligence*, 23(2):123–150, 2009.
- [33] W. Dodd. *The design of procedural, semantic, and episodic memory systems for a cognitive robot*. PhD thesis, Vanderbilt University, 2005.
- [34] W. Dodd and R. Gutierrez. The role of episodic memory and emotion in a cognitive robot. In *IEEE International Workshop on Robots and Human Interactive Communication*, pages 692–697, 2005.
- [35] A. Egges, S. Kshirsagar, and N. Magnenat-Thalmann. Generic personality and emotion simulation for conversational agents. *Computer Animation and Virtual Worlds*, 15(1):1–13, 2004.

-
- [36] R. El Kaliouby and P. Robinson. Generalization of a vision-based computational model of mind-reading. In *Affective computing and intelligent interaction*, pages 582–589, 2005.
- [37] M. S. El-Nasr, J. Yen, and T. R. Ioerger. FLAME—fuzzy logic adaptive model of emotions. *Autonomous Agents and Multi-agent systems*, 3(3):219–257, 2000.
- [38] H. C. Ellis and B. A. Moore. Mood and memory. *Handbook of Cognition and Emotion*, pages 193–210, 1999.
- [39] S. Erk, M. Kiefer, J. Grothe, A. P. Wunderlich, M. Spitzer, and H. Walter. Emotional context modulates subsequent memory effect. *Neuroimage*, 18(2):439–447, 2003.
- [40] Z. Faltersack, B. Burns, A. Nuxoll, and T. L. Crenshaw. Ziggurat: Steps toward a general episodic memory. In *AAAI Fall Symposium: Advances in Cognitive Systems*, pages 106–111, 2011.
- [41] D. François, S. Powell, and K. Dautenhahn. A long-term study of children with autism playing with a robotic pet: Taking inspirations from non-directive play therapy to encourage children’s proactivity and initiative-taking. *Interaction Studies*, 10(3):324–373, 2009.
- [42] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.
- [43] N. H. Frijda. Emotion, cognitive structure, and action tendency. *Cognition and emotion*, 1(2):115–143, 1987.
- [44] N. H. Frijda. Varieties of affect: Emotions and episodes, moods, and sentiments. *The Nature of Emotions: Fundamental Questions*, pages 197–202, 1994.
- [45] P. Gebhard. ALMA: a layered model of affect. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 29–36. ACM, 2005.

-
- [46] P. Gebhard, T. Baur, I. Damian, G. Mehlmann, J. Wagner, and E. André. Exploring interaction strategies for virtual characters to induce stress in simulated job interviews. In *Proceedings of the international conference on Autonomous agents and multi-agent systems*, pages 661–668, 2014.
- [47] P. Gebhard and K. H. Kipp. Are computer-generated emotions and moods plausible to humans? In *Intelligent Virtual Agents*, pages 343–356. Springer, 2006.
- [48] J. Gemrot, R. Kadlec, M. Bída, O. Burkert, R. Píbil, J. Havlíček, L. Zemčák, J. Šimlovič, R. Vansa, M. Štolba, et al. Pogamut 3 can assist developers in building ai (not only) for their videogame agents. In *Agents for Games and Simulations*, pages 1–15. Springer, 2009.
- [49] R. Gockley, A. Bruce, J. Forlizzi, M. Michalowski, A. Mundell, S. Rosenthal, B. Sellner, R. Simmons, K. Snipes, A. C. Schultz, et al. Designing robots for long-term social interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1338–1343, 2005.
- [50] L. R. Goldberg. An alternative “description of personality”: the big-five factor structure. *Journal of Personality and Social Psychology*, 59(6):1216, 1990.
- [51] P. F. Gomes, C. Martinho, and A. Paiva. I’ve been here before!: location and appraisal in memory retrieval. In *The 10th International Conference on Autonomous Agents and Multiagent Systems*, pages 1039–1046, 2011.
- [52] J. Gratch and S. Marsella. A domain-independent framework for modeling emotion. *Cognitive Systems Research*, 5(4):269–306, 2004.
- [53] D. Griol and Z. Callejas. An architecture to develop multimodal educative applications with chatbots. *International Journal of Advanced Robotic Systems*, 10, 2013.

-
- [54] J. J. Gross. Antecedent-and response-focused emotion regulation: divergent consequences for experience, expression, and physiology. *Journal of Personality and Social Psychology*, 74(1):224, 1998.
- [55] R. Guenther. Mood and memory. *Memory in Context: Context in Memory*, GM Davies and DM Thomson, Editors, pages 57–79, 1988.
- [56] H. Gunes and M. Piccardi. Bi-modal emotion recognition from expressive face and body gestures. *Journal of Network and Computer Applications*, 30(4):1334–1345, 2007.
- [57] Y. Guoliang, W. Zhiliang, W. Guojiang, and C. Fengjun. Affective computing model based on emotional psychology. In *Advances in Natural Computation*, pages 251–260. Springer, 2006.
- [58] A. Hartholt, D. Traum, S. C. Marsella, A. Shapiro, G. Stratou, A. Leuski, L.-P. Morency, and J. Gratch. All Together Now: Introducing the Virtual Human Toolkit. In *13th International Conference on Intelligent Virtual Agents*, Edinburgh, UK, Aug. 2013.
- [59] D. O. Hebb. The organization of behavior, 1949.
- [60] F. Heuer and D. Reisberg. Emotion, arousal, and memory for detail. *The Handbook of Emotion and Memory: Research and Theory*, pages 151–180, 1992.
- [61] W. C. Ho, K. Dautenhahn, and C. L. Nehaniv. Autobiographic agents in dynamic virtual environments-performance comparison for different memory control architectures. In *IEEE Congress on Evolutionary Computation*, volume 1, pages 573–580, 2005.
- [62] M. W. Howard and M. J. Kahana. When does semantic similarity help episodic retrieval? *Journal of Memory and Language*, 46(1):85–98, 2002.

-
- [63] A. M. Isen and B. Means. The influence of positive affect on decision-making strategy. *Social cognition*, 2(1):18–31, 1983.
- [64] J. S. Janowsky, A. P. Shimamura, and L. R. Squire. Source memory impairment in patients with frontal lobe lesions. *Neuropsychologia*, 27(8):1043–1056, 1989.
- [65] S. Jockel, D. Westhoff, and J. Zhang. EPIROME—A novel framework to investigate high-level episodic robot memory. In *IEEE International Conference on Robotics and Biomimetics*, pages 1075–1080, 2007.
- [66] R. Kadlec and C. Brom. DyBaNeM: Bayesian framework for episodic memory modelling. In *The 12th International Conference on Cognitive Modelling, ICCM*, 2013.
- [67] R. Kadlec, M. Čermák, Z. Behan, and C. Brom. Generating corpora of activities of daily living and towards measuring the corporas complexity. In *Cognitive Agents for Virtual Environments*, pages 149–166. Springer, 2012.
- [68] Z. Kasap and N. Magnenat-Thalmann. Towards episodic memory-based long-term affective interaction with a human-like robot. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 452–457, 2010.
- [69] Z. Kasap and N. Magnenat-Thalmann. Building long-term relationships with virtual and robotic characters: the role of remembering. *The Visual Computer*, 28(1):87–97, 2012.
- [70] Z. Kasap, M. B. Moussa, P. Chaudhuri, and N. Magnenat-Thalmann. Making them remember - emotional virtual characters with memory. *IEEE Computer Graphics and Applications*, 29(2):20–29, 2009.
- [71] S. Kshirsagar. A multilayer personality model. In *Proceedings of the 2nd International Symposium on Smart Graphics*, pages 107–115. ACM, 2002.
- [72] P. J. Lang. The emotion probe: Studies of motivation and attention. *American psychologist*, 50(5):372, 1995.

-
- [73] R. S. Lazarus. *Emotion and adaptation*. Oxford University Press New York, 1991.
- [74] J. LeDoux and J. R. Bemporad. The emotional brain. *Journal of the American Academy of Psychoanalysis*, 25(3):525–528, 1997.
- [75] C. M. Lee and S. S. Narayanan. Toward detecting emotions in spoken dialogs. *IEEE Transactions on Speech and Audio Processing*, 13(2):293–303, 2005.
- [76] I. Leite, C. Martinho, and A. Paiva. Social robots for long-term interaction: a survey. *International Journal of Social Robotics*, 5(2):291–308, 2013.
- [77] I. Leite, C. Martinho, A. Pereira, and A. Paiva. As time goes by: Long-term evaluation of social presence in robotic companions. In *IEEE International Symposium on Robot and Human Interactive Communication*, pages 669–674, 2009.
- [78] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.
- [79] H. Liang, J. Yuan, and D. Thalmann. Parsing the hand in depth images. *IEEE Transactions on Multimedia*, 16(5):1241–1253, 2014.
- [80] G. Liu, S. Choudhary, J. Zhang, and N. Magnenat-Thalmann. Let’s keep in touch online: a facebook aware virtual human interface. *The Visual Computer*, 29(9):871–881, 2013.
- [81] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.
- [82] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

-
- [83] F. Mairesse and M. A. Walker. Towards personality-based user adaptation: psychologically informed stylistic language generation. *User Modeling and User-Adapted Interaction*, 20(3):227–278, 2010.
- [84] S. Marsella and J. Gratch. EMA: A computational model of appraisal dynamics. *European Meeting on Cybernetics and Systems Research*, 2006.
- [85] S. Marsella, J. Gratch, and P. Petta. Computational models of emotion. *A Blueprint for Affective Computing-A Sourcebook and Manual*, pages 21–46, 2010.
- [86] R. R. McCrae and P. T. Costa. Validation of the five-factor model of personality across instruments and observers. *Journal of personality and social psychology*, 52(1):81, 1987.
- [87] B. McDaniel, S. DMello, B. King, P. Chipman, K. Tapp, and A. Graesser. Facial features for affective state detection in learning environments. In *Proceedings of the 29th Annual Cognitive Science Society*, pages 467–472, 2007.
- [88] A. Mehrabian. Measures of individual differences in temperament. *Educational and Psychological Measurement*, 38(4):1105–1117, 1978.
- [89] A. Mehrabian. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, 14(4):261–292, 1996.
- [90] T. Mikolov and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013.
- [91] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751, 2013.
- [92] R. Mises and H. Pollaczek-Geiringer. Praktische verfahren der gleichungsauflösung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 9(1):58–77, 1929.

-
- [93] W. N. Morris and P. P. Schnurr. *Mood: The Frame of Mind*. Springer-Verlag Publishing, 1989.
- [94] M. B. Moussa and N. Magnenat-Thalmann. Toward socially responsible agents: integrating attachment and learning in emotional decision-making. *Computer Animation and Virtual Worlds*, 2013.
- [95] S. T. Mueller and R. M. Shiffrin. REM II: A model of the developmental co-evolution of episodic memory and semantic knowledge. In *International Conference on Learning and Development (ICDL)*, 2006.
- [96] A. Newell. *Unified theories of cognition*, volume 187. Harvard University Press, 1994.
- [97] J. E. Newhagen and B. Reeves. The evening’s bad news: Effects of compelling negative television news images on memory. *Journal of Communication*, 42(2):25–41, 1992.
- [98] A. M. Nuxoll and J. E. Laird. Extending cognitive architecture with episodic memory. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 1560, 2007.
- [99] H. Orozco, F. Ramos, M. Ramos, and D. Thalmann. A fuzzy model to update the affective state of virtual humans: An approach based on personality. In *International Conference on Cyberworlds (CW)*, pages 406–413, 2010.
- [100] A. Ortony. On making believable emotional agents believable. *R. P. Trapple, P. (Ed.), Emotions in humans and artefacts*, pages 189–211, 2002.
- [101] A. Ortony, G. Clore, and A. Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, 1988.

-
- [102] A. Papangelis, N. Kouroupas, V. Karkaletsis, and F. Makedon. An adaptive dialogue system with online dialogue policy learning. In *7th Hellenic Conference on Artificial Intelligence*, pages 323–330. Springer, 2012.
- [103] H.-S. Park and C.-H. Jun. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2):3336–3341, 2009.
- [104] S. Pulman, J. Boye, M. Cavazza, C. Smith, and R. De La Cámara. ‘how was your day?’. In *Proceedings of the 2010 Workshop on Companionable Dialogue Systems*, pages 37–42. Association for Computational Linguistics, 2010.
- [105] F. Rabe and I. Wachsmuth. An event metric and an episode metric for a virtual guide. In *Proceedings of the 5th International Conference on Agents and Artificial Intelligence*, volume 2, 2013.
- [106] R. Reisenzein. Pleasure-arousal theory and the intensity of emotions. *Journal of Personality and Social Psychology*, 67(3):525, 1994.
- [107] J. Ren, X. Jiang, and J. Yuan. Learning lbp structure by maximizing the conditional mutual information. *Pattern Recognition*, 2015.
- [108] M. A. Salichs and M. Malfaz. A new approach to modeling emotions and their use on a decision-making system for artificial agents. *IEEE Transactions on Affective Computing*, 3(1):56–68, 2012.
- [109] T. Salter, K. Dautenhahn, and R. Bockhorst. Robots moving out of the laboratory-detecting interaction levels and human contact in noisy school environments. In *IEEE International Workshop on Robot and Human Interactive Communication*, pages 563–568, 2004.
- [110] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

-
- [111] R. Santos, G. Marreiros, C. Ramos, J. Neves, and J. Bulas-Cruz. Personality, emotion and mood simulation in decision making. *Proceedings of 14th Portuguese Conference on Artificial Intelligence. EPIA. Aveiro, October 12-15, 2009.*
- [112] K. R. Scherer. Appraisal considered as a process of multilevel sequential checking. *Appraisal processes in emotion: Theory, methods, research*, 92:120, 2001.
- [113] K. R. Scherer and H. Ellgring. Multimodal expression of emotion: Affect programs or componential appraisal patterns? *Emotion*, 7(1):158, 2007.
- [114] M. A. M. Shaikh, H. Prendinger, and M. Ishizuka. Sentiment assessment of text by analyzing linguistic features and contextual valence assignment. *Applied Artificial Intelligence*, 22(6):558–601, 2008.
- [115] F. Shu and A.-H. Tan. A biologically-inspired affective model based on cognitive situational appraisal. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2012.
- [116] G. Sieber and B. Krenn. Towards an episodic memory for companion dialogue. In *Intelligent Virtual Agents*, pages 322–328. Springer, 2010.
- [117] A. Staller and P. Petta. Introducing emotions into the computational study of social norms: A first evaluation. *Journal of artificial societies and social simulation*, 4(1):27–60, 2001.
- [118] B. R. Steunebrink, M. Dastani, and J.-J. C. Meyer. The OCC model revisited. In *Proceeding of the 4th Workshop on Emotion and Computing*, 2009.
- [119] M. Steyvers, T. L. Griffiths, and S. Dennis. Probabilistic inference in human semantic memory. *Trends in Cognitive Sciences*, 10(7):327–334, 2006.
- [120] C. Strapparava and R. Mihalcea. Learning to identify emotions in text. In *Proceedings of the ACM symposium on Applied computing*, pages 1556–1560, 2008.

-
- [121] D. Tecuci and B. W. Porter. An episodic based approach to complex event processing. In *AAAI Spring Symposium: Intelligent Event Processing*, pages 86–91, 2009.
- [122] D. G. Tecuci and B. W. Porter. *A generic memory module for events*, volume 68. 2007.
- [123] D. G. Tecuci and B. W. Porter. A generic memory module for events. In *Proceedings to the 20th Florida Artificial Intelligence Research Society Conference (FLAIRS 20)*, Key West, FL, 2007.
- [124] A. Tellegen. *The MMPI-2 Restructured Clinical (RC) Scales: Development, Validation, and Interpretation*. University of Minnesota Press, 2003.
- [125] E. Thorson and M. Friestad. The effects of emotion on episodic memory for television commercials. *Cognitive and Affective Responses to Advertising*, pages 305–325, 1989.
- [126] E. Tulving. Episodic and semantic memory. *Organization of memory*, pages 381–402, 1972.
- [127] E. Tulving. Precis of elements of episodic memory. *Behavioral and Brain Sciences*, 7(2):223–268, 1984.
- [128] M. Turunen, J. Hakulinen, O. Ståhl, B. Gambäck, P. Hansen, M. C. R. Gancedo, R. S. De La Cámara, C. Smith, D. Charlton, and M. Cavazza. Multimodal and mobile conversational health and fitness companions. *Computer Speech and Language*, 25(2):192–209, 2011.
- [129] A. van der Heide and G. Trivino. Simulating emotional personality in human computer interfaces. In *IEEE International Conference on Fuzzy Systems (FUZZ)*, pages 1–7, 2010.

-
- [130] J. D. Velásquez. When robots weep: emotional memories and decision-making. In *Fifteenth National Conference on Artificial Intelligence, Madison, WI.*, pages 70–75, 1998.
- [131] H. Vilhjálmsson, N. Cantelmo, J. Cassell, N. E. Chafai, M. Kipp, S. Kopp, M. Mancini, S. Marsella, A. N. Marshall, C. Pelachaud, et al. The behavior markup language: Recent developments and challenges. In *International Workshop on Intelligent Virtual Agents*, pages 99–111. Springer, 2007.
- [132] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967.
- [133] K. Wada and T. Shibata. Robot therapy in a care house-its sociopsychological and physiological effects on the residents. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3966–3971, 2006.
- [134] J. Wagner, F. Lingenfelser, T. Baur, I. Damian, F. Kistler, and E. André. The social signal interpretation (ssi) framework: multimodal signal processing and recognition in real-time. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 831–834. ACM, 2013.
- [135] D. Wang and A.-H. Tan. Creating autonomous adaptive agents in a real-time first-person shooter computer game. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(2):123–138, 2015.
- [136] W. Wang, B. Subagdja, A.-H. Tan, and J. A. Starzyk. Neural modeling of episodic memory: Encoding, retrieval, and forgetting. *IEEE transactions on neural networks and learning systems*, 23(10):1574–1586, 2012.
- [137] Y. Wilks, R. Catizone, S. Worgan, A. Dingli, R. Moore, D. Field, and W. Cheng. A prototype for a conversational companion for reminiscing about images. *Computer Speech and Language*, 25(2):140–157, 2011.

- [138] R. Wood, P. Baxter, and T. Belpaeme. A review of long-term memory in natural and synthetic systems. *Adaptive Behavior*, 20(2):81–103, 2012.
- [139] Y. Xiao, H. Liang, J. Yuan, and D. Thalmann. Body movement analysis and recognition. In *Context Aware Human-Robot and Human-Agent Interaction*, pages 31–53. Springer, 2016.
- [140] J. Zhang, J. Zheng, and N. Magnenat-Thalmann. PCMD: personality-characterized mood dynamics model toward personalized virtual characters. *Computer Animation and Virtual Worlds*, 26(3-4):237–245, 2015.
- [141] Z. Zhang, A. Beck, and N. Magnenat-Thalmann. Human-like behavior generation based on head-arms model for robot tracking external targets and body parts. *IEEE Transactions on Cybernetics*, 45(8):1390–1400, 2015.

Author's Publications

1. J. Zhang, J. Zheng and N. Magnenat Thalmann, "Modeling emotions and moods in an affective system for virtual human and social robots". In 25th international conference on computer animation and social agents (CASA), Singapore, 2012
2. J. Zhang, J. Zheng and N. Magnenat Thalmann, "PCMD: Personality-Characterized Mood Dynamics Model Towards Personalized Virtual Characters", *Computer Animation and Virtual Worlds*, 26(3-4): 237-245, 2015
3. J. Zhang, J. Zheng and N. Magnenat Thalmann, "Modeling Personality, Mood, and Emotions", *Context Aware Human-Robot and Human-Agent Interaction*, Springer International Publishing, 211-236, 2015
4. J. Zhang, N. Magnenat Thalmann and J. Zheng, "Combining memory and emotion with dialogue for social companions, a review", *International Conference on Computer Animation and Social Agents*, 2016.
5. G. Liu, S. Choudhary, J. Zhang, and N. Magnenat-Thalmann, "Let's keep in touch online: a Facebook aware virtual human interface", *The Visual Computer*, 1-11, 2013.
6. J. Zhang, J. Zheng and N. Magnenat Thalmann. "Episodic Memory for Long-term Companion Free Dialogue in Natural Language". Submitted to ACM Transactions on Interactive Intelligent Systems (TIIS).
7. J. Zhang, J. Zheng. "Mixed-Correlation-Analysis Based Episodic Memory for Intelligent Agents". To be Submitted

Appendix A: Questionnaire of User Study for PCMD Model

Questionnaire

Thank you very much for attending this user study.

Please fill in your background information below. All of your response will be kept strictly confidential.

What is your name? _____

How old are you? _____

Please indicate your gender:

_____ Male _____ Female

How do you know about the virtual human or social robot?

Nothing 1 2 3 4 5 6 7 Very much

How do you know about affect computing?

Nothing 1 2 3 4 5 6 7 Very much

The questions in the following pages ask about the experience you just had.

There is no right or wrong answers. Please simply give your first impressions and answer all of the questions as accurately as possible

Figure 7.1: Page 1 of questionnaire

Interaction Videos

Please watch the four videos in order. The videos are about the **emotional interactions** between user and social robot.

The user will **make equal positive or negative comments** on the robot. The word and the order of making comments are kept same for all videos.

The **baseline** is that we suppose a **neutral** robot should be happy for good comments and be angry for negative comments.

Please compare the personalities of the robot shown in the following videos with this baseline.

After watching each video, please answer the questions about the **personalities** of the robot.

Figure 7.2: Page 2 of questionnaire

Please watch the first video and answer the following questions.

A. Social Robot 1

1. Compared with negative comments, the social robot is more sensitive to positive comments

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

2. Compared with positive comments, the social robot is more sensitive to negative comments

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

3. The personality of the social robot can be easily recognized.

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

4. The personality of the social robot is

Negative 1 2 3 4 5 6 7 Positive

Please watch the second video and answer the following questions.

B. Social Robot 2

1. Compared with negative comments, the social robot is more sensitive to positive comments

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

2. Compared with positive comments, the social robot is more sensitive to negative comments

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

3. The personality of the social robot can be easily recognized.

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

4. The personality of the social robot is

Negative 1 2 3 4 5 6 7 Positive

Figure 7.3: Page 3 of questionnaire

Please watch the third video and answer the following questions.

C. Social Robot 3

1. Compared with negative comments, the social robot is more sensitive to positive comments

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

2. Compared with positive comments, the social robot is more sensitive to negative comments

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

3. The personality of the social robot can be easily recognized.

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

4. The personality of the social robot is

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

4. The personality of the social robot is

Negative 1 2 3 4 5 6 7 Positive

Please watch the fourth video and answer the following questions.

D. Social Robot 4

1. Compared with negative comments, the social robot is more sensitive to positive comments

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

2. Compared with positive comments, the social robot is more sensitive to negative comments

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

3. The personality of the social robot can be easily recognized.

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

4. The personality of the social robot is

Negative 1 2 3 4 5 6 7 Positive

Figure 7.4: Page 4 of questionnaire

Appendix B: Questionnaire of User Study for MCAEM Model

Instructions

The aim of the user study is to validate the effects of episodic memory in long-term virtual human (VH)-user interactions in natural language.

Participants will interact with two kinds of virtual human (named Nicole), one with memory and one without memory. The order is arranged randomly.

The interactions are separated into two sessions.

In the first session, participants will interact with Nicole as new users. Nicole will try to discuss with you about your nationality, current work, research topic and hobby. Of course, you can ask any questions about Nicole too (e.g. "What's your name"). After Nicole gathering all the wanted information, she will end the interaction automatically.

Participants can communicate with Nicole in free language. But please give direct answers immediately after Nicole's questions. The answers can include those negative ones, such as "I don't know", "I have no idea".

In the second session, participants will meet Nicole again. Participants are required to ask Nicole questions about the previous discourses. For example, "What is my nationality?", "Do you know my research topic?". After a few rounds, Nicole will ask your permission to ask more questions about you. You can end interactions any time by saying "goodbye" to Nicole.

After the two sessions, please complete the prepared questionnaire.

Figure 7.5: Page 1 of questionnaire

Questionnaire

Thank you very much for attending this user study.

Please fill in your background information below. All of your response will be kept strictly confidential.

What is your name? _____

How old are you? _____

Please indicate your gender:

_____ Male _____ Female

How do you know about the virtual human or social robot?

Nothing 1 2 3 4 5 6 7 Very much

How do you know about episodic memory?

Nothing 1 2 3 4 5 6 7 Very much

The questions in the following pages ask about the experience you just had.

There is no right or wrong answers. Please simply give your first impressions.

Figure 7.6: Page 2 of questionnaire

Please answer the following questions.

1. How involving was your interaction with Nicole?
 Not at all 1 2 3 4 5 6 7 Very much
2. How engaging was the conversation?
 Not at all 1 2 3 4 5 6 7 Very much
3. To what extent did you feel mentally immersed in the experience?
 Not at all 1 2 3 4 5 6 7 Very much
4. To what extent did you experience a sensation of reality?
 Not at all 1 2 3 4 5 6 7 Very much
5. I was so involved in my interaction with Nicole that I lost track of time.
 Not at all 1 2 3 4 5 6 7 Very much
6. How often did you want to or did you make eye-contact with Nicole?
 Not at all 1 2 3 4 5 6 7 Very much
7. How often did you make a sound out loud (e.g. laugh, speak) in response to Nicole?
 Not at all 1 2 3 4 5 6 7 Very much
8. How often did you smile or make an expression in response to Nicole?
 Not at all 1 2 3 4 5 6 7 Very much
9. The dialogue between the Companion and me felt natural
 Not at all 1 2 3 4 5 6 7 Very much
10. I thought the dialogue was appropriate
 Not at all 1 2 3 4 5 6 7 Very much

Figure 7.7: Page 3 of questionnaire

11. Nicole can understand what I mean.

Not at all 1 2 3 4 5 6 7 Very much

12. Nicole has a good understanding of the conversation context.

Not at all 1 2 3 4 5 6 7 Very much

13. The responses of Nicole are easy to understand.

Not at all 1 2 3 4 5 6 7 Very much

14. The logic of the dialogue is coherent.

Not at all 1 2 3 4 5 6 7 Very much

15. I thought Nicole remembered me when I met her again.

Not at all 1 2 3 4 5 6 7 Very much

16. I thought Nicole talked with me in aware of our past conversation

Not at all 1 2 3 4 5 6 7 Very much

17. Nicole can remember almost everything I have talked.

Not at all 1 2 3 4 5 6 7 Very much

18. Nicole has good memory of past dialogues.

Not at all 1 2 3 4 5 6 7 Very much

19. Nicole can recall related knowledge from other people.

Not at all 1 2 3 4 5 6 7 Very much

20. The recalled memories are coherent to current context.

Not at all 1 2 3 4 5 6 7 Very much

21. I think the knowledge of Nicole grows with the conversation.

Not at all 1 2 3 4 5 6 7 Very much

Figure 7.8: Page 4 of questionnaire

22. Nicole has good capability of learning.

Not at all 1 2 3 4 5 6 7 Very much

23. I think Nicole would be helpful for me as a virtual assistant in the future.

Not at all 1 2 3 4 5 6 7 Very much

24. I think Nicole is good at reminding me past events.

Not at all 1 2 3 4 5 6 7 Very much

25. Over time I think I would build up a relationship with the Companion

Not at all 1 2 3 4 5 6 7 Very much

26. I hope to interact with Nicole again in the future.

Not at all 1 2 3 4 5 6 7 Very much

Figure 7.9: Page 5 of questionnaire