

國立臺灣大學電機資訊學院電機工程學系

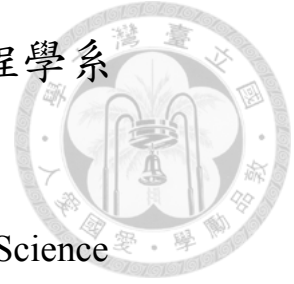
碩士論文

Department of Electrical Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis



自傳式記憶輔助型對話系統

Dialogue System Assisted by Autobiographical Memory

蕭羽庭

Yu-Ting Hsiao

指導教授：傅立成博士

Advisor: Li-Chen Fu, Ph.D.

中華民國 108 年 7 月

July, 2019



誌謝

能完成這項研究承蒙許多人的幫助與鼓勵，首先我想感謝我的指導教授傅立成老師。在研究的道路上總是適時地給予我許多激勵與啟發，讓我能夠不斷地思考並改進，不斷地挖掘自己的潛能，面對各種挑戰時永不放棄。同時也感謝口試委員蘇木春老師、項天瑞老師、黃從仁老師以及李宏毅老師，老師們所提供的專業建議讓次研究能更於完善，在此對所有的老師表示真摯的感謝。

在碩士生涯中，實驗室的成員情誼將是值得且難忘的回憶。謝謝所有學長姐的經驗傳承於教導，碩一新生時有士桓學長的幫助，再來有宗澤學長的在研究上的帶領，以及秉蒼學長、Vicente、Cesar、竣棠和行之的相助。他們讓我在碩一的時候學到許多做研究的技巧與精神，這些都是支持我繼續走下去的支柱。還要感謝兩年來明理學長、安陞學長、欣怡學姊在求學的過程中給予的各種學術指導。謝謝助理懿萱、小寧與郁璇在行政事務莫大的協助。

謝謝一同努力的啟維、天時、少宏、雅慧、恩宇、昱文、宇閔與逸霖的相互打氣與交流，感謝學弟其安、瀟越給予的幫助與關懷。也要感謝更多未列名於此的朋友們，你們的友情支援讓我的碩士生涯順利結束。

特別感謝我最親愛的家人一直以來無私的付出與包容，讓我能夠無後顧之憂的進行求學與研究。最後再次感謝所有關心著我的人。

蕭羽庭 謹致於
國立臺灣大學電機工程學系
尖端控制實驗室
2019年7月29日



摘要

老人照護是目前人們很關注的一個社會議題。將機器人導入老人照護的領域有助於降低老人照護的人力需求。其中，社交型機器人能夠通過與老人的交流為老人帶來心靈上的支持。

在本篇論文中，目標是能夠開發一個能夠和老人聊他們的故事的機器人對話系統。此系統的關鍵在於將一個能進行科學計算的記憶模型引入到對話系統中，讓機器人擁有記住人類個人資訊的能力並且能夠依照相關的個人資訊對人進行回覆。為了了解人類的語言並且從中抽取出合適的資訊，本篇論文中提出了一個自然語言理解的模組，由基礎的自然語言處理工具、模板及規則結合而成。在對話管理模組的控制下，從人類語言中提取的資訊會被送到自傳式記憶模組，一個負責儲存人類記憶並且能夠抽取相關記憶內容的模組。之後，抽取出的相關記憶內容會被送到 Skills 模組，生成機器人的回覆。最後，生成的機器人回覆會經由自然語言生成模組展現給人們。展現機器人的回覆的同時從記憶模組中抽取出的記憶內容也會被展現出來。

系統的評估經由設計好的幾個實驗。實驗的結果顯示本篇論文提出的系統能夠利用人類的語言抽取記憶資訊並從記憶的知識庫提取相關的記憶內容。並且此系統能夠將記憶內容融合到機器人的生成語言中，以此和人盡心交流。此揭示了機器人能夠對老人提供心理層面照護的可能性，以及和老人建立陪伴關係的可能性。

關鍵字：對話系統、自傳式記憶、自然語言理解、核心對話模組



Abstract

Elderly care is a social issue that humans are focus on. It is helpful to introduce robots into the field of elderly care, which alleviates the requirement of manpower of taking care of senior citizens. Social robots can especially provide elders with mental support through communication.

In this thesis, the aim is to develop a dialogue system for robots which can chat with elders about their own stories. And the key is to embed a computational memory model into the dialogue system so as to give robots the ability to memorize human's personal information as well as to generate relevant responses. To understand human utterances and extract appropriate information, a Natural Language Understanding module is proposed, consisting of rules, patterns and typical natural language processing tools. Then, under the control of designed Dialogue Management module, the extracted information are sent to Autobiographical Memory module, where memory of humans is stored and memory items are retrieved. After that, the retrieved memory items are sent to Skills module, from which various robot's responses are generated. Finally, the generated responses are presented to humans through a Natural Language Generation module, which shows both robot's utterances and depicts retrieved memory items.

The system is evaluated through several well designed experiments. The results show that the system proposed is able to utilize human's utterances to extract memory information and retrieve memory items from knowledge base. Moreover it is also given the ability to generate responses embedded

with memory items to chat with humans, which reveals the potential for offering mental supports to elders so as to build the companionship.

Keywords: Dialogue System, Autobiographical Memory, Natural Language Understanding, Core Chat





Contents

誌謝	i
摘要	ii
Abstract	iii
Contents	v
List of Figures	viii
List of Tables	x
1 Introduction	1
1.1 Background	1
1.2 Motivation	2
1.3 Related Work	4
1.3.1 Memory Model	5
1.3.2 Dialogue System	12
1.4 Objective and Contribution	18
1.5 Thesis Organization	19
2 Preliminary	20
2.1 Relational Database	20
2.1.1 Structured Query Language	22
2.2 Artificial Intelligence Markup Language	23

2.3	Part-of-speech Tagging	25
2.4	Dependency Parsing	26
2.5	Recurrent Neural Network	27
2.5.1	Long Short-Term Memory	29
2.5.2	Gated Recurrent Unit	30
2.6	Sequence to Sequence	31
2.6.1	Seq2Seq with Attention	31
3	Methodology	34
3.1	System Overview	34
3.2	Autobiographical Memory	35
3.2.1	Memory Knowledge Base	36
3.2.2	Memory Database Management System	43
3.3	Natural Language Understanding	51
3.3.1	Event Extraction	52
3.3.2	Sentiment Analysis	62
3.4	Dialogue Management	62
3.4.1	Global State Tracker	62
3.4.2	Global Dialogue Policy	63
3.5	Skills	64
3.5.1	Core Chat	64
3.5.2	Script Bots	65
3.6	Natural Language Generation	68
4	Evaluation	70
4.1	Evaluation of Natural Language Understanding	70
4.1.1	Data Description	71
4.1.2	Participants	71
4.1.3	Procedure	71
4.1.4	Results and Discussion	72



4.2	Evaluation of Autobiographical Memory	75
4.2.1	Data Description	75
4.2.2	Procedure	76
4.2.3	Results and Discussion	77
4.3	Evaluation of Core Chat	80
4.3.1	Data Description	80
4.3.2	Parameters	81
4.3.3	Results and Discussion	81
5	Conclusion	86
	References	89

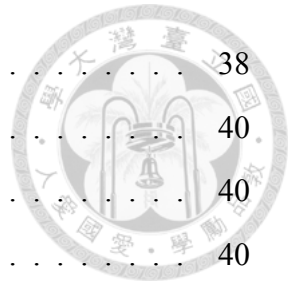




List of Figures

1.1	Types of memory. ¹	5
1.2	Structure of autobiographical memory knowledge base in Self-Memory System [1].	7
1.3	An example of conversation between Eliza and a human [2].	14
1.4	An example of Seq2Seq framework used for response generation [3].	15
1.5	A typical architecture of a task-oriented dialogue-state system [4].	16
1.6	The architecture of Sounding Board [5].	17
2.1	An example of RDB.	20
2.2	An example of chatting with Mitsuku online.	25
2.3	An example of POS tagging in simplified Chinese using LTP, which is an open-source tools for Natural Language Processing.	26
2.4	An example of dependency parsing in simplified Chinese using LTP.	27
2.5	The working principle of unfolded RNN.	28
2.6	The working principle of unfolded BRNN.	28
2.7	The internal structure of an LSTM cell.	30
2.8	The internal structure of an GRU cell.	31
2.9	An illustration of Seq2Seq.	32
2.10	The working principle of an attention mechanism.	32
3.1	The architecture of overall system.	34
3.2	The architecture of Memory KB.	37
3.3	The relational schema of theme table.	37

3.4	The relational schema of <code>lifetime</code> table.	38
3.5	The relational schema of <code>human</code> table.	40
3.6	The relational schema of <code>location</code> table.	40
3.7	The relational schema of <code>vn</code> table.	40
3.8	The relational schema of <code>episodic</code> table.	42
3.9	The workflow of Event Extraction Module.	52
3.10	The architecture of Core Chat module (Adapted from [6]).	64
4.1	The distribution of collected postings over 11 categories.	71
4.2	An example of the testing pair for the evaluation of NLU module.	73
4.3	The distribution of collected postings over 6 categories.	75
4.4	An example of the posting and its summary.	76
4.5	The experiment result of <i>Recall</i> for <i>Threshold</i> of 3.	79
4.6	The training loss of Core Chat module.	82
4.7	The validation loss of Core Chat module.	82





List of Tables

1.1	Ten characteristics of episodic memory [1].	9
3.1	Tags of POS tagging from LTP. (Adapted from ² .)	55
3.2	Rules for extracting items of human.	56
3.3	Rules for extracting items of location.	57
3.4	Rules for extracting items of activity & object.	58
3.5	Rules for extracting items of fact.	60
3.6	Rules for extracting items of age.	61
4.1	Questions contained in the questionnaire for each posting.	72
4.2	The <i>Precision</i> , <i>Recall</i> and F_1 <i>Score</i> of NLU module of different categories.	73
4.3	The <i>Precision</i> , <i>Recall</i> and F_1 <i>Score</i> of NLU module of all postings.	74
4.4	Results of Autobiographical Memory module under low memory density.	77
4.5	Results of Autobiographical Memory module under high memory density.	78
4.6	Examples of data used to train, valid and test Core Chat module.	81
4.7	The perplexity of Core Chat module compared with Seq2SeqAtt.	83
4.8	The result of human judgement for Core Chat module.	84
4.9	Examples of responses generated by Core Chat module.	85



Chapter 1

Introduction

1.1 Background

In this century, with the development of science and technology, human beings tend to live longer. This phenomenon, accompanied with the low birth rate, results in a large change to the distribution of age, leading to an increasing proportion of elderly population. Consequently, some new focuses related to elderly population are raised. One of the worldwide issues is the elderly care, which aims to fulfill the special needs and requirements from senior citizens.

Plenty of solutions have been raised to help elders, such as the establishment of nursing homes, policy of long-term care, or provision of daily care from social workers. With the help of these strategies, the elders are empowered to maintain their functionality, autonomy, and quality of life [7]. However, these health care systems are going to encounter a large shortage of qualified workers, due to the gradually increasing aging population [8] [9]. Therefore, some new issues rise. One issue is that even though the elders have already moved to nursing homes, the attention paid to each individual is little. It's

hard for only a few caregivers to comprehensively take care of a number of elders. Another issue is that the limited number of nursing homes makes plenty of elders who need assistance remain in their houses, despite some of them may prefer to live in their own places. Unfortunately, there is a ceiling of the number of social workers who can service elders house-to-house.

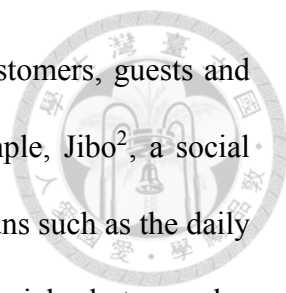
Facing the shortage of manpower in elderly care, the significant role of science and technology is highlighted [10]. Some solutions have already turned up, such as telehealth, telemonitoring, or telecare [11]. Among these solutions, robots also play an important role. Assistive robots can be roughly divided in to two categories [12]. One category is to provide physical assistive technology such as smart wheelchairs or artificial limbs, which can barely communicate with elders. The other category is to offer psychological help or companionship. For example, in the early 21st century, AIBO [13], the first robot proposed for entertainment, is introduced to the world.

It is commonly believed that introducing robots into elderly care can not only figure out the problem of shortage of manpower but also reduce the expense of elderly care. Most of all, robots have beneficial effects on elders from both physical and psychological aspects. Therefore, in order to bring a better life to the elder population, the progress in developing robots is immediately necessary.

1.2 Motivation

Recently, social robots have received significant attention. They are developed to assist humans in several occasions. Some of them work in public. For example, Pepper¹, one of the most famous robots, shows up in banks, stores, or hotels, standing at the front

¹<https://www.softbankrobotics.com/us/pepper>



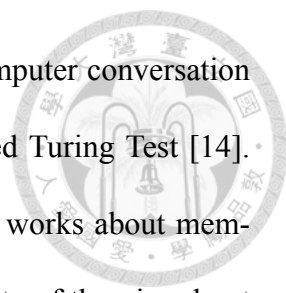
door or in front of the information desk, providing guidance for customers, guests and visitors. Social robots can also serve in private sectors. For example, Jibo², a social robot designed for the home, can deliver several information to humans such as the daily news and the weather forecast, or remind humans of the schedule. Social robots are also introduced to institutes providing elderly care for senior citizens. Pepper has been used in some nursing homes to cheer elders up by dancing. And, Miro³ can not only keep company but also watch over their health status.

As has been described, various social robots have been introduced to the public. Their skills are dependent on their roles. When considering social robots as companions, one of their most essential capabilities is to chat. Robots mimic caregivers, providing mental support to elders through communicating. However, a large gap still exists between human-human conversation and human-robot conversation. One main factor is the ability to memorize. Humans can know each other through communication, because they are able to remember interests, hates, and past experiences of each other. And based on these knowledges, humans tend to talk more with each other, which later makes them friends. Robots, to the best of our knowledge, haven't been gifted with human's ability to memorize things. Without the ability to memorize, no matter how many times has an elder talked with a robot, the robot still treats the elder as a stranger and keeps asking the elder similar questions. If robots can only provide general conversations, it's barely possible that the elders can treat robots as companions, which limits the effectiveness of mental supports that robots can bring to elderly care.

To go beyond the limitation of human-robot conversation, one solution is to enable robots to memorize. In other words, it demands an integration of robot's dialogue system

²<https://www.jibo.com/>

³<http://consequentialrobotics.com/miro>



with memory model. Researchers have been working on human-computer conversation since last century, and some of the systems have successfully passed Turing Test [14]. In contrast to the abundant research done on dialogue systems, few works about memory model can be found in the field of engineering, even though plenty of theories about memory exist in the field of psychology. Therefore, the integration of robot's dialogue system with memory may face several challenges. Firstly, computational memory models are not popular topics in the field of engineering, that is, it's barely possible to find an existing memory model which is suitable for integrating into the robot's dialogue systems. Secondly, a well-designed Natural Language Understanding (NLU) module is needed in order to retrieve memory entities from human's utterances. However, most of the dialogue systems are designed without memory, which means that their NLU modules do not function with the ability to retrieve memory entities, so that few NLU modules can be consulted. Thirdly, the generation of robot's response embedded with memory items is challenging. Even though there exist abundant of works trying to control the style of generated machine's utterances through various methods, it's still hard to utilize memory items to control robot's utterances.

1.3 Related Work

In order to enable robots to memorize things, it's important to introduce researches on memory models in the field of psychology, which helps to figure out how memory is stored in humans' brains. Along with theories of memory models, it takes some advantages to review works related to memory in the field of engineering. Later on, existing dialogue systems are reviewed in this thesis. In fact, there are several ways which can be used to divide existing dialogue systems, including task-oriented or non-task-oriented, rule-based

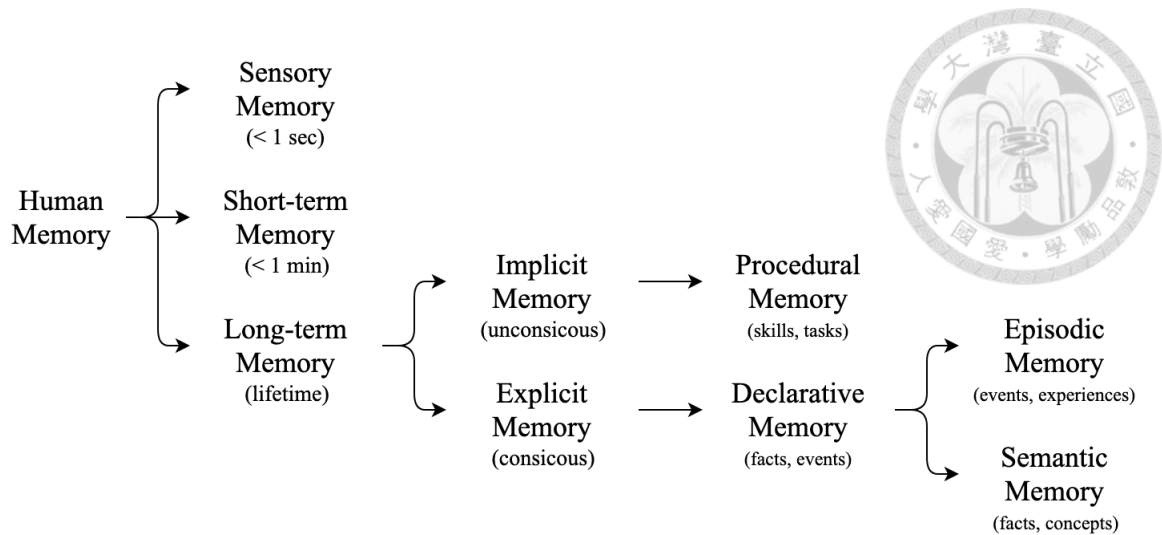


Figure 1.1: Types of memory.⁴

or corpus-based, and dialogue-state or non-dialogue-state.

1.3.1 Memory Model

Memory relates to how information is processed in human's brain. There are three main functions which lead to the process, namely, encoding of memory, storage of memory, and retrieval of memory [15]. It is vital for humans to preserve their memory, since it represents their past. Without the past, a person can not learn or develop language, relationship, or personal identity [16].

Memory can be divided into several categories, as illustrated in Figure 1.1. Among these categories, the declarative memory, which is the memory of facts and events and is a subclass of long-term memory, is worth noticing in this thesis. It can be further separated into two sub-categories, semantic memory and episodic memory. Semantic memory refers to general factual knowledge about the world, which is independent of individuals⁵. In contrast, episodic memory is related to individual memory of personal experiences and

⁴Adapted from <http://www.human-memory.net/types.html>

⁵For example, the apple is a kind of fruits and the lettuce is a kind of vegetables.

specific objects, people and events which happened at particular places and at particular time⁶.

After further studies, the boundary between semantic memory and episodic memory can be somewhat vague [17]. For example, a person remembered that he/she took a holiday to Italy last year, which may involve both semantic memory and episodic memory. *Italy* and *last year* can be classified as episodic memory, whereas the meaning of *holiday* can be viewed as semantic memory. From this point of view, semantic memory and episodic memory are not fully separated but in an intertwined relationship. Therefore, in this thesis, autobiographical memory is adopted to represent human's memory, which is a memory model holding both semantic and episodic knowledge.

• *Autobiographical Memory from Psychology*

Autobiographical memory is fundamentally significant to human beings, which is a recollection of episodes from one's life, containing both semantic knowledge and episodic knowledge.

Since Tulving raised his theory about autobiographical memory [18], researches on autobiographical memory has lasted over 40 years in various subareas of psychology, such as in cognitive, clinical or neuropsychological areas. However, most of their findings are isolated, due to high complexity of autobiographical memory. It was not organized into a framework until the Self-Memory System was proposed by Conway and Pleydell-Pearce in 2000 [19]. The Self-Memory System is mainly separated into two parts. One is autobiographical knowledge, where episodes are recollected. The other one is the working-self, which sets personal goals and manages the control process like encoding of memories and

⁶For example, on the day of one's commencement, his/her parents took a photo with him/her in front of the school gate with a bunch of flowers in hands.

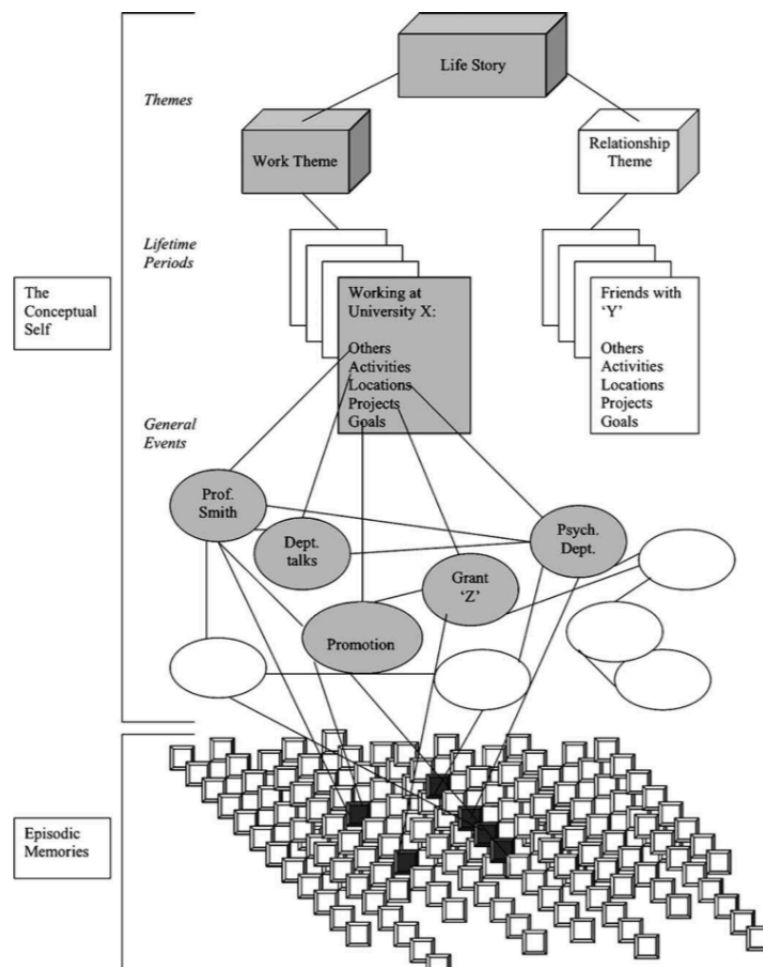


Figure 1.2: Structure of autobiographical memory knowledge base in Self-Memory System [1].

the accessibility of memories.

After the Self-Memory System was proposed, several findings tried to illustrate this system. In 2005, Conway did a systematic review of the system [1]. Despite that the main components of the Self-Memory System were left the same, some details were elaborated. The latest version of autobiographical memory knowledge base is shown in Figure 1.2.

Information, or memories, in autobiographical memory knowledge base is stored hierarchically. The knowledge base can be roughly divided into two layers, the conceptual self and episodic memories. The top layer aims to store semantic knowledge whereas the lower layer aims to store episodic knowledge. Taking one step further, to fully discover

the mechanism of the autobiographical memory knowledge base, a four-layer structure is introduced, namely, **themes**, **lifetime periods**, **general events** and **episodic memories**.

Themes is the most abstract level, which divides the memory knowledge base into several categories. Each item in **Themes** can access particular items in *Lifetime periods*. For example, in Figure 1.2, *Work Theme* links to specific items in *Lifetime periods* which represent the conceptual knowledge about work, whereas *Relationship Theme* links to other particular items which are related to friends.

Lifetime periods is a layer between **Themes** and **General Events**. It is used to access items in *General Events*. An item in **Lifetime periods** contains a set of knowledge that are common to that period, such as goals, others (humans), locations, activities, or evaluations. As illustrated in Figure 1.2, an item called *Working at University X* utilize its attributes to link to items in **General Events**. For example, *Others of Working at University X*, which stores information about humans, is linked to *Prof. Smith*, and *Goals* of that is linked to *Promotion*.

General Events includes repeated or categoric events, extended events, and mini-histories, which are cues to access specific memories in **Episodic Memories**. For example, *Prof. Smith* in *General Events* may link to *The first meeting with labmates* or *Attending conference in Taipei in the first year*, which are items with specific details stored in **Episodic Memories**.

Compared to **Episodic Memories**, definitions of items or their attributes are much clearer in **Themes**, **Lifetime periods** and **General Events**. Even though several characteristics of episodic memory have been proposed, as shown in Table 1.1, few works discuss about how episodic memory is represented or what categories or attributes can be used to describe episodic memory. In a study of episodic memory, Conway [20] conducted a sim-



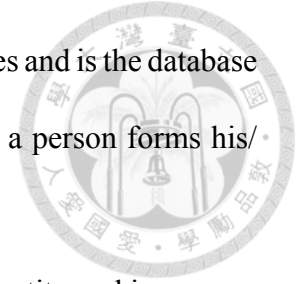
Table 1.1: Ten characteristics of episodic memory [1].

-
- 1 Retain summary records of sensory-perceptual-conceptual-affective processing derived from working memory.
 - 2 Retain patterns of activation or inhibition over long periods.
 - 3 They are predominately represented in the form of images.
 - 4 Represent short time slices, determined by changes in goal-processing.
 - 5 Represented roughly in their order of occurrence.
 - 6 They are only retained in a durable form if they become linked to conceptual autobiographical knowledge, otherwise, rapidly forgotten.
 - 7 Their main function is to provide a short-term record of progress in current goal processing.
 - 8 They are recollectively experienced when accessed.
 - 9 When included as part of an autobiographical memory construction they provide specificity.
 - 10 Neuroanatomically they may be represented in brain regions separate from other autobiographical knowledge networks.
-

ple experiment, in which the categories of descriptions of episodic memory is introduced. In the experiment, participants were asked to recall what they could remember from their homes to the laboratory, including thoughts and feelings. After participants listed their descriptions, two judges developed a coding scheme to classify the statements in descriptions of memories. The coding scheme contained several categories: action, thought, feeling, location and fact, which is adopted in this thesis to store episodic memories.

In [17], Williams discussed about three functions of autobiographical memory to show how important autobiographical memory is to human beings. The first one is the directive function, which shows that autobiographical memory uses memories of past events to guide current and future behavior including problem-solving. The second one is the social function. The social function provides material for conversations and helps social interactions, which is the progress of making friends with each other. The third function defines how autobiographical memory is related to the self, and it is viewed as the most

important function. Autobiographical memory stores personal histories and is the database from which a person is formed. Through autobiographical memory a person forms his/her personal identity.



To sum up, autobiographical memory is essential to a person's identity and is a representation of one's history. The conceptual model of autobiographical memory knowledge base in Self-Memory System can obtain both semantic and episodic memories. Therefore, it's reasonable for robots to understand someone through his/her autobiographical memory. It is believed that with the help of autobiographical memory, robots can bring better companionship to elders.

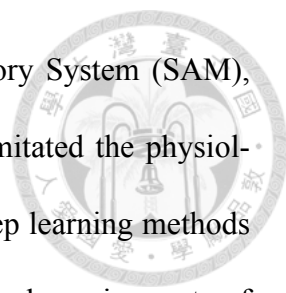
• *Autobiographical Memory in Engineering*

With the development of social robots, a few researchers have tried to introduce autobiographical memory into their robot systems. Some aim to help robots understand the world and others try to consolidate the relationship between humans and robots.

In order to let robots understand the world, Pointeau [21] proposed an autobiographical memory system on the iCub⁷, a humanoid robot, so that the robot could accumulate its experiences while interacting with humans and formed composite knowledge in its mind. Their autobiographical memory system followed Tulving's theorem [18], which consisted of two parts, Episodic-Like Memory and Semantic memory. The memory model, implemented using PostgreSQL⁸, tried to store inputs from humans, *i.e.*, names, actions, time, actions from robots, and pertinent information of context, space or time of the world. They applied their work to the development of robot's knowledge of spatial locations, relations between objects, and temporal relations between events.

⁷<http://www.icub.org/>

⁸<https://www.postgresql.org/>

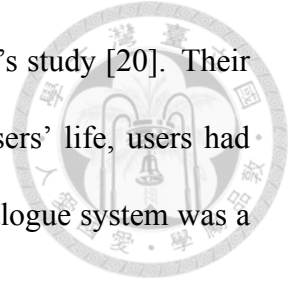


Some researchers worked on Synthetic Autobiographical Memory System (SAM), which not only required robots to understand the world, but also imitated the physiology of the brain. In [22], a hippocampus architecture assisted by deep learning methods is proposed as a modern SAM, which could better fulfilled functional requirements of autobiographical memory. The work [23] implemented SAM on the iCub in order to let robots understand the surrounding environment and improve robot's perception. The iCub gathered information including human faces, human arm movements, and gestures from multiple sensors. After learning, it is tested by perception of faces, actions, and gestures.

Besides enabling robots to understand the world, the integration of robots and memory models is also used for consolidation of the human-robot interactions. In [24], they tried to integrate a computational autobiographical memory with intelligent virtual agents so as to create coherent life stories for virtual agents, which aims to achieve long-term believability. The computational autobiographical memory utilized three parts, Abstract, Narrative, and Evaluation, to store information of object, location, human, and so on. Another work [25] proposed a memory model in order to let robots achieve long-term companion. The proposed memory model enabled robots to remember events which are significant or relevant to themselves or users. The structure of the memory model provided not only space for storage but also management of goals in order to reason robot's actions. Robots embedded with memory models are expected to become more believable companions.

Some researchers also put effort into the integration of dialogue systems and memory models. In [26], they claimed that continuous communication between humans is promoted by sharing experiences, and therefore, MAY, a memory-gathering conversational virtual companion is developed. The autobiographical memory model embedded in MAY, similar

to the memory knowledge base in this thesis, was based on Conway's study [20]. Their experiment showed that even providing little acquaintance about users' life, users had more positive attitude towards human-robot relationship. MAY's dialogue system was a modification of A.L.I.C.E.⁹, which is a chatbot relying on scripts.



1.3.2 Dialogue System

Researchers have worked on dialogue systems for several decades, endowing machines with the ability to talk with humans. The ultimate goal is to let human-robot conversation as similar as possible to human-human conversation. So far, existing dialogue systems can be divided into several categories, task-oriented or non-task-oriented, rule-based or corpus-based, and dialogue-state or non-dialogue-state.

• *Task-oriented versus Non-task-oriented Dialogue System*

Task-oriented dialogue systems aim to solve particular tasks and are required to finish their tasks within few interactions with humans. Several remarkable examples can be listed, such as Siri¹⁰, Alexa¹¹, and Google Assistant¹². These dialogue systems obey commands from human and can manage schedules, make phone calls or search for information online. Moreover, some of them can answer questions or work as consultants, like DoNotPay, which is a virtual lawyer helping people with incorrect parking fines, insurance issues, and so on.

In contrast, non-task-oriented dialogue systems are designed to keep human company, that is, they are not asked to complete specific tasks, but should be able to generate

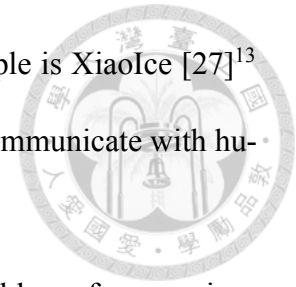
⁹<https://www.chatbots.org/chatbot/a.l.i.c.e/>

¹⁰<https://www.apple.com/tw/siri/>

¹¹<https://www.alexa.com/>

¹²<https://assistant.google.com/>

conversations with human in boundless domain. One famous example is XiaoIce [27]¹³ developed by Microsoft, which utilizes a great number of skills to communicate with human.



The proposed dialogue system in this thesis aims to tackle the problem of companionship, so it is classified as a non-task-oriented dialogue system, which is able to generate responses embedded with memory items in boundless topics, that is, the proposed system listens to humans talking about their experiences and then give related responses. Therefore, the following review will focus on non-task-oriented dialogue systems, or they can be called chatbots.

• *Rule-based versus Corpus-based Dialogue System*

In earlier research on chatbot, most of the systems are rule-based. Eliza [2], is known as the first chatbot which is a simulation of a Rogerian psychotherapist. Through hand-crafted scripts and methods of pattern matching, Eliza can accept textual inputs and then generate corresponding responses. Figure 1.3 shows a conversation between Eliza and a human being, where capitalized sentences indicate utterances generated by Eliza. Even though the conversation shown in Figure 1.3 seems to be quite fluent, due to the limitation on Eliza's knowledge, it can only chat with humans within a constrained domain.

Another world-famous chatbot is Alice¹⁴, which was developed in 2009. Alice uses an Artificial Intelligence Markup Language (AIML) to let its users customize their chatbots. Since AIML tags allow chatbots to call a pattern matcher recursively, the scripts needed by chatbots are simplified. Alice was the winner of the Loebner Prize¹⁵ in 2000, 2001, and 2004. Recently, still based on AIML, Mitsuku¹⁶ has won the Loebner Prize for four

¹³<https://www.msxiaobing.com/>

¹⁴<https://home.pandorabots.com/home.html>

¹⁵<http://www.aisb.org.uk/events/loebner-prize>

¹⁶<https://www.pandorabots.com/mitsuku/>

Men are all alike.
IN WHAT WAY?
They're always bugging us about something or other.
CAN YOU THINK OF A SPECIFIC EXAMPLE?
Well, my boyfriend made me come here.
YOUR BOYFRIEND MADE YOU COME HERE
He says I'm depressed much of the time.
I AM SORRY TO HEAR YOU ARE DEPRESSED



Figure 1.3: An example of conversation between Eliza and a human [2].

times (in 2013, 2016, 2017, 2018). Due to the power of AIML, some of the skills in this thesis are built upon it.

One main weakness of rule-based dialogue systems is the limited amount of scripts. Fortunately, with the advancement of research on dialogue systems, several corpus-based dialogue systems have been proposed to deal with existing issues. Corpus-based chatbots learn to respond to humans from data of human-human conversations, human-machine conversations or even non-dialogue documents. They can be separated into two categories, IR-based (information retrieval based) and ML-based (machine learning based). IR-based chatbots use several similarity functions to extract proper responses from their databases. Most commonly used similarity functions are cosine similarities between word features, such as the cosine similarity between word embeddings. Several IR-based chatbots [28] [29] [30] have already succeeded in response generation. The response generated by IR-based chatbots guarantees its performance on grammar, but it's lack of adaptability on sentences, since sentences are retrieved from databases.

In contrast to IR-based chatbots, ML-based, or generation-based chatbots can not guarantee the performance on grammar and sometimes may even generate some nonsenses,

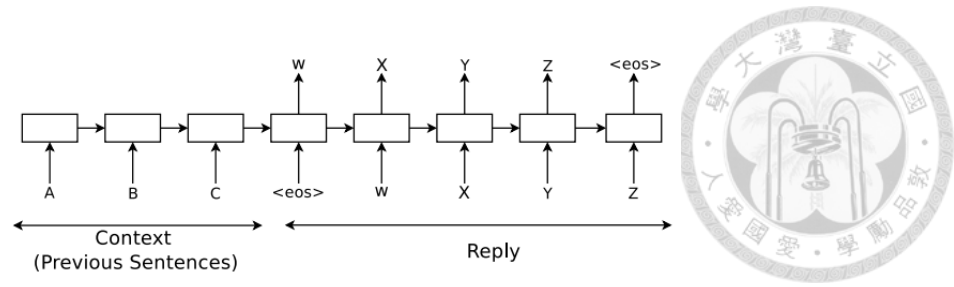


Figure 1.4: An example of Seq2Seq framework used for response generation [3].

but they can generate more human-like sentences due to their adaptability on the construction of sentences. ML-based chatbots are mostly constructed in an encoder-decoder framework, which is also known as Sequence-to-sequence (Seq2Seq). Seq2Seq was first developed to deal with machine translation tasks and was later adopted in response generation [31] [3] [32]. An example of chatbot with Seq2Seq framework is shown in Figure 1.4.

Compared to machine translation tasks, it's much more difficult to tackle tasks of response generation. For one input text, it is likely to have various plausible responses. Therefore, a lot of researchers keep working in this field, proposing several modified versions of Seq2Seq. For example, in order to prevent chatbots from always producing repetitive and dull responses like "I don't know", Li [33] proposed using Maximum Mutual Information (MMI) instead of maximum log-likelihood as the objective function so that chatbots could generate more diverse and interesting responses. Some other works also aim to let chatbots generate more informative responses. The work [34] incorporated copying mechanism into Seq2Seq so as to replicate certain segments in input queries into output responses. Another work [6] also tried to bring external knowledge into the generated sentences. They firstly obtained topic words from Latent Dirichlet Allocation (LDA) model, and then a joint attention mechanism is performed to affect the decoding process. Extra probability was added to topic words in order to let generated words bias upon

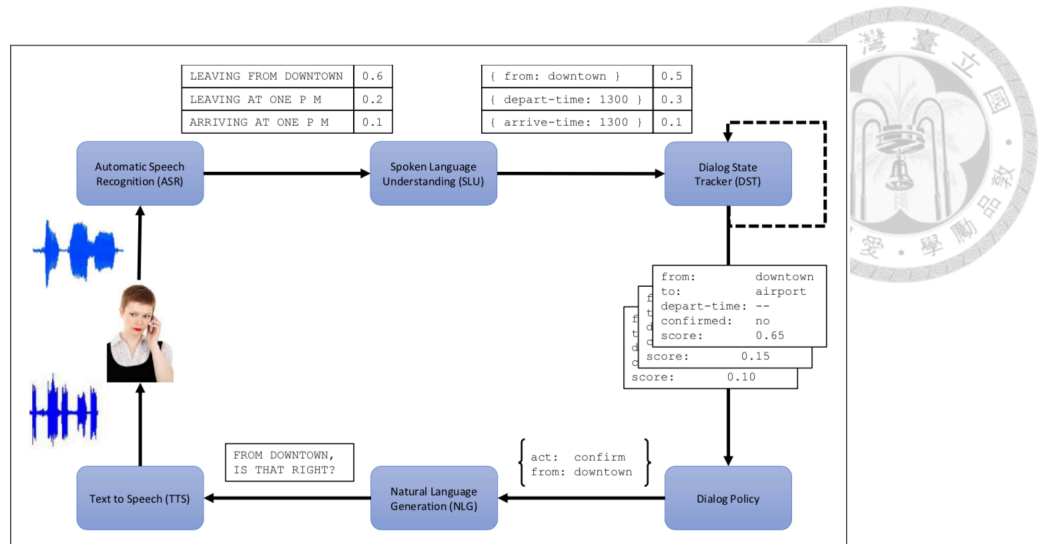


Figure 1.5: A typical architecture of a task-oriented dialogue-state system [4].

them. Their experimental results showed profit on incorporating external knowledge into generated sentences. In this thesis, based upon their model framework, with some modifications, we try to let our proposed system generate robot's responses embedded with memory items.

• *Dialogue-state versus Non-dialogue-state Dialogue System*

Either IR-based chatbots or ML-based chatbots mentioned above are both classified as non-dialogue-state dialogue system. However, when taking commercial products into considerations, a more complex system seems to be much more suitable.

Most of the chatbots on the market are developed under complicated structure, such as Siri, Alexa or XiaoIce. They obtain plenty of skills to give responses to users. A typical dialogue-state system, as shown in Figure 1.5, may help to explain working mechanisms in their complicated systems. Firstly, the inputs from users is transferred into texts through Automatic Speech Recognition (ASR) module. Next, a Spoken Language Understanding (SLU) module is used to extract necessary information from input texts. The extracted information updates dialogue states in Dialogue State Tracker (DST) module. After that,

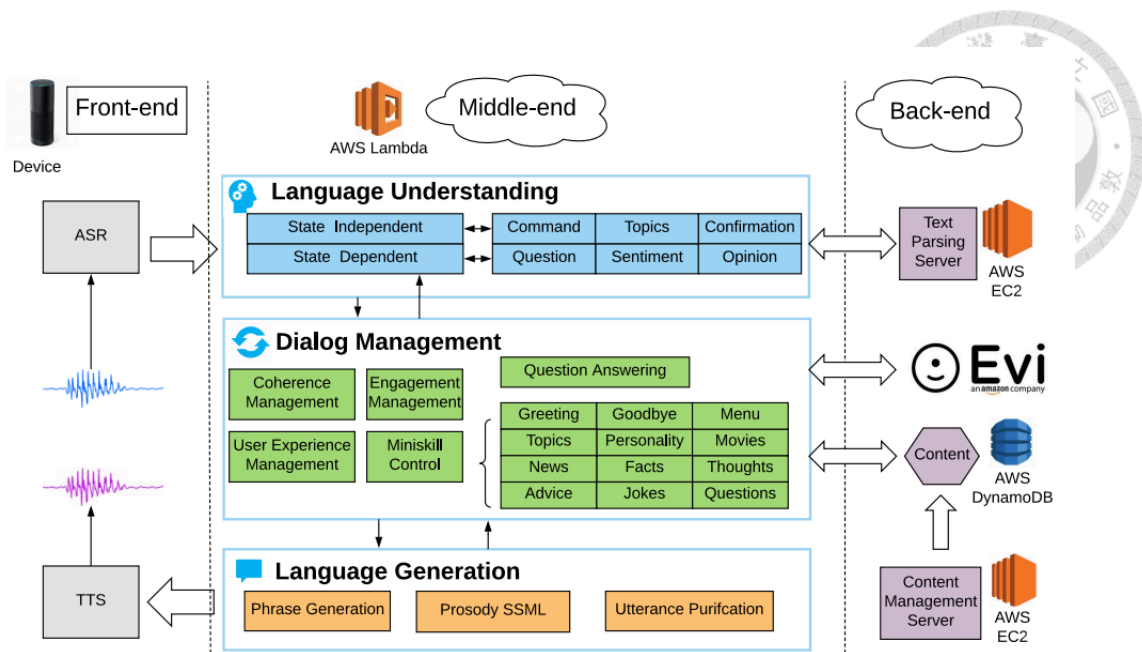


Figure 1.6: The architecture of Sounding Board [5].

the Dialogue Policy (DP) module makes decisions on dialogue actions, *i.e.*, *confirm*, based on DST. Finally, through Natural Language Generation (NLG) module, information is transferred into sentences and is spoken out by Text to Speech (TTS) module. Modules in this architecture can be divided into individual research areas, and can be designed based on requirements of systems.

Sounding Board [5], a chatbot proposed by University of Washington in Alexa Prize¹⁷, also adopted the architecture of dialogue-state system with some modification. The architecture of Sounding Board is shown in Figure 1.6. NLU module is similar to SLU, which detects human's intents and extracts useful information. Dialogue Manager (DM) module can be viewed as a combination of DST and DP. DM controls several skills to gain knowledge, and through NLG the knowledge is transferred into sentences. As the winner of Alexa Prize, the advantage of using complicated systems is verified.

In this thesis, the architecture of the proposed system is similar to that of Sounding

¹⁷<https://developer.amazon.com/alexaprize>

Board but with redefined modules to meet the requirement of utilizing memory items to chat with elders.



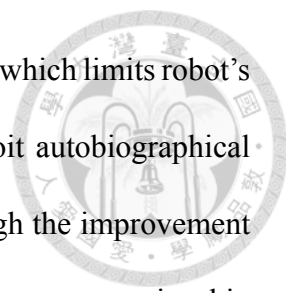
1.4 Objective and Contribution

For the purpose of assisting elderly care with robots, the system aims to provide better human-robot communication through the integration of computational autobiographical memory model and dialogue system. Storing elders experiences in robot's mind, it enables robots to know what an elder is favorite of or what has happened to an elder before, which helps robots to provide elders with better mental support through conversations. Three objectives are addressed in this thesis.

The first objective is to build a computational autobiographical memory knowledge base as well as its database management system (DBMS). In this work, a conceptual framework of autobiographical memory called Self-Memory System from the field of Psychology is adopted. And a probabilistic-based retrieval method is designed as a part of its DBMS.

The second objective is to develop a suitable NLU module in order to extract proper information from human utterances. Since items stored in our autobiographical memory knowledge base are specific terms, a well-designed module is needed to extract essential information without noise.

Finally, several skills are proposed to help generate robot's responses. These skills are required to generate both general responses as well as responses embedded with memory items. Besides, a procedure is needed to identify human identity so as to load human memory from knowledge base. To make conversations complete and better, some script bots are designed to offer helps.



Previous works on conversational agents are mostly memory-less, which limits robot's ability on elderly care. In this thesis, our system attempts to exploit autobiographical memory knowledge base to assist human-robot conversation. Through the improvement of human-robot conversation, it is believed that robots can build closer companionship with elders and bring more benefits to elderly care.

1.5 Thesis Organization

The rest of this thesis is organized as follows.

In Chapter 2, some tools and basic knowledge are introduced. Firstly, Structured Query Language (SQL) and Artificial Intelligence Markup Language (AIML) are presented. SQL is used for storing and managing autobiographical memory knowledge base and AIML is the basic of the proposed script bots. Then, basic knowledge of deep learning on generation of sequences is briefly introduced, which is the main techniques of sentences generation.

In Chapter 3, the proposed dialogue system with autobiographical memory is presented. Firstly, an overview of the proposed system is given. Next, the structure of autobiographical memory knowledge base used in this thesis is introduced, so as its memory database management system. Then components of our dialogue system are described one by one, including NLU module, DM module, Skills, and NLG module.

In Chapter 4, the evaluation of the proposed system is divided into three parts, the evaluation of NLU module, the evaluation of autobiographical memory module and the evaluation of chatting module.

At the end of this thesis, a conclusion is conducted in Chapter 5.



Chapter 2

Preliminary

2.1 Relational Database

In order to maintain a large amount of data, database and database management system (DBMS) are developed. Database is where data is stored whereas DBMS is a system which enables users to manipulate and control access to the database. Relational database (RDB) is an extension of database proposed by E. F. Codd in 1970 [35], and correspondingly its DBMS is called relational database management system (RDBMS), where the data stored are relational, referring to Figure 2.1.

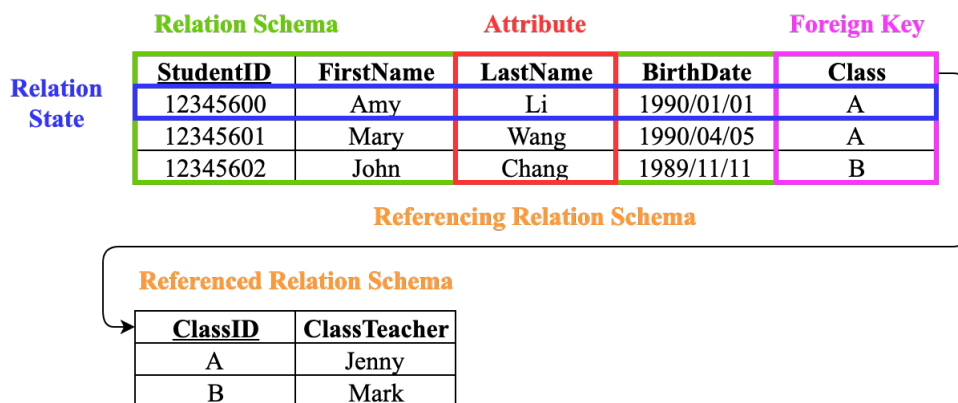
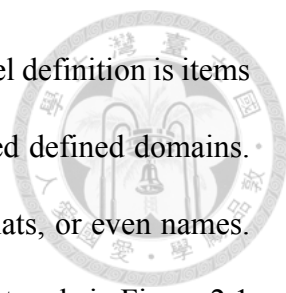


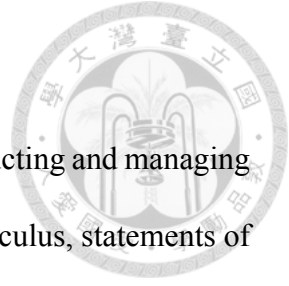
Figure 2.1: An example of RDB.



A three-layer definition is used to introduce RDB. The lowest-level definition is items stored in the database. Values of items in the RDB should not exceed defined domains. Here, domains are sets of atomic values, which have types and formats, or even names. The medium-level definition is relation schema, which is the green rectangle in Figure 2.1 and can be simply viewed as a table. A relation schema has a relation name and a list of attributes, which all have attribute names. In a relation schema, a row is a relation state and a column is an attribute. The top-level definition is of database schema, which is the definition of all relation schemas in this database.

In order to make databases more meaningful and more useful, several model constraints are proposed. One of the main constraints is on primary keys. Primary keys are used to identify relation states within a relation schema. Thus, they can not be replicated. Moreover, primary keys can contain more than one attribute. It is common to indicate primary keys by underlining attribute names. For example, in Figure 2.1, *StudentID* and *ClassID* are primary keys with respect to their tables. Another essential constraint in RDB is on foreign keys. Foreign keys help to link relation schemes. Foreign keys in a referencing relation schema tie to primary keys in a referenced relation schema, as indicated in Figure 2.1. Foreign keys can be duplicate in referencing relation schemas, but need to be unique in referenced schemas. In addition, a relation state in referencing relation schemas must refer to an existing relation state in referenced schemas.

RDB and RDBMS is commonly used to store information of financial records, personal data and manufacturing information. Therefore, in this thesis, RDB and RDBMS are adopted to store personal memories heard from elders and retrieve relevant entities to response to elders.



2.1.1 Structured Query Language

Structured Query Language (SQL) is a useful language for constructing and managing a RDB and RDBMS. Based upon relational algebra and relational calculus, statements of SQL can be simply classified as data definition language (DDL) and data manipulation language (DML).

DDL defines database schemas, which is able to create, alter or drop database objects, *i.e.*, *tables*. An example of creating tables in Figure 2.1 is shown as follows. Firstly, the type (*i.e.*, TABLE) of the relation schema is declared and a name (*i.e.*, *class*) is given. Then, each attribute is declared by a name (*i.e.*, *ClassID*) and a domain (*i.e.*, *VARCHAR(10)*) to constrain its values. Finally, constraints on tables need to be specified, *i.e.*, to indicate that *ClassID* is the primary key, or to indicate that *Class* is a foreign key refer to *ClassID* in the referenced table *class*.

```
CREATE TABLE class (  
    ClassID VARCHAR(10),  
    ClassTeacher VARCHAR(30),  
    PRIMARY KEY(ClassID));  
  
CREATE TABLE student (  
    StudentID INT,  
    FirstName VARCHAR(30),  
    LastName VARCHAR(30),  
    BirthDate DATE,  
    Class VARCHAR(10),  
    PRIMARY KEY(StudentID),
```

```
FOREIGN KEY(Class) REFERENCES class(ClassID);
```

DML is used for inserting, updating, and deleting data in a RDB. Besides, it can perform selecting to retrieve data from RDB. An example of inserting data into tables in Figure 2.1 is shown as follows. Several terms need to be specified, the table to insert (*i.e.*, `class`), the attributes to insert (*i.e.*, `ClassID`), and the values to insert (*i.e.*, `('A', 'Jenny')`).

```
INSERT INTO class(ClassID, ClassTeacher)
VALUES ('A', 'Jenny'), ('B', 'Mark');
```

In this thesis, our RDB and RDBMS are constructed and are manipulated using MySQL¹, which is an open-source to implement SQL.

2.2 Artificial Intelligence Markup Language

Artificial Intelligence Markup Language (AIML), developed by Richard Wallace in 2003, is an XML dialect aiming to help people build their own rule-based conversational system. Following its rules, conversational agents match people's inputs to patterns and generate responses based on individual templates. For example, as shown below, if a user says 'Hello I am Jenny', then the response given will be 'Hi Jenny'. The wildcard (*) in the pattern is to extract information from input sentences and place the extracted information at `<star/>`, so that personal information can be used in machine's responses.

```
<category>
```

```
  <pattern>Hello I am *</pattern>
```

¹<https://www.mysql.com/>



```
<template>Hi <star/></template>  
</category>
```

The power of AIML is that it is able to call itself recursively. In the example as follows, if the input from users is ‘How can I call you’ , the <srai> tag will map the input to ‘What is your name’ and then the template used to respond will be ‘My name is Pepper’ . This ability enables machines to generate various responses within short scripts.

```
<category>  
  <pattern>What is your name</pattern>  
  <template>My name is Pepper</template>  
</category>  
  
<category>  
  <pattern>How can I call you</pattern>  
  <template>  
    <srai>What is your name</srai>  
  </template>  
</category>
```

AIML has already achieved remarkable success on human-computer conversations. Alicebot (Artificial Linguistic Internet Computer Entity) is developed based on AIML, which has won the Loebner Prize² for three times (in 2000, 2001 and 2004) and has been released as an open source. Another outstanding chatbot is Mitsuku³, which is a four-time winner of the Loebner Prize (in 2013, 2016, 2017, and 2018). Mitsuku can be viewed as an offspring of Alicebot, which inherits all scripts from Alice and plenty of additional

²The Loebner Prize is a format of Turing test which is held annually to award prizes to the most human-like computer programs.

³<https://www.pandorabots.com/mitsuku/>

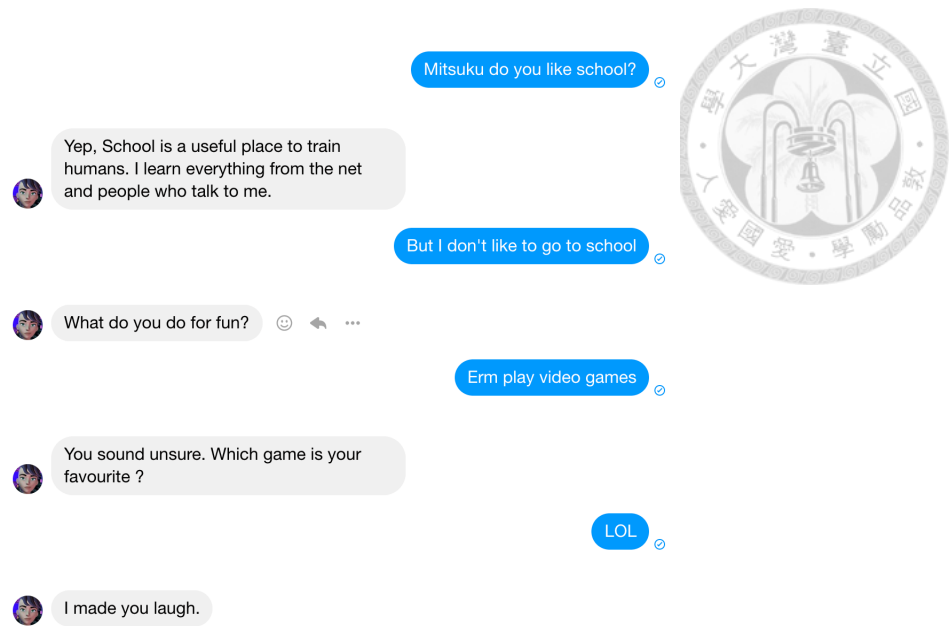


Figure 2.2: An example of chatting with Mitsuku online.

scripts. It is kept maintaining until now and is available on several social platforms. Users can talk with Mitsuku on diverse topics, as shown in Figure 2.2.

In this thesis, AIML is adopted to build rule-based chatting modules which makes conversations more complete.

2.3 Part-of-speech Tagging

Part-of-speech Tagging (POS tagging), which is also called grammatical tagging, is a process of giving marks to words with respect to definitions of words and contexts. It can also be simply viewed as marking words with grammatical tags, such as noun, verb, or adjective. POS tagging is the first step of various tasks in the field of computational linguistics, for instance, syntactic parsing, which is to build grammatical trees for sentences, and information extraction, which aims to extract correlated info from texts. An example of POS tagging in simplified Chinese is shown in Figure 2.3. In the example, “元芳 (YuanFang)” is tagged as “nh” , which indicates human name, “你 (you)” and “怎

元芳 你 怎么 看
nh r r v



Figure 2.3: An example of POS tagging in simplified Chinese using LTP, which is an open-source tools for Natural Language Processing.

麼 (how)” are tagged as “r” , which means pronoun, and “看 (see)” is tagged as “v” , which means verb.

Several well-known open-source tools for Chinese POS tagging have been developed, such as NLTK⁴, Jieba⁵, THULAC⁶, LTP⁷, and pkuseg⁸. Some of their tags are in common, such as noun, verb and adjective, while some of those are different. For example, some words are tagged as *an* by Jieba, which indicates that these words can be both noun and adjective, but LTP doesn’t offer this tag.

In this thesis, in order to extract people’s memory from their utterance, POS tagging from LTP is adopted, which helps to accomplish the task of information extraction.

2.4 Dependency Parsing

Dependency parsing is a parsing that links each lexical items in a sentence with directed relations. Despite of word-order information, dependency parsing represents the information which is necessary for the parse. A sentence through dependency parsing will form a dependency tree. A dependency tree need to satisfy several constraints. Firstly, each tree should have a single root node without incoming arcs. Secondly, each nodes in the tree, except for the root node, should have exactly one incoming arc. Thirdly, in each

⁴<http://www.nltk.org/>

⁵<https://github.com/fxsjy/jieba>

⁶<http://thulac.thunlp.org/>

⁷<https://github.com/HIT-SCIR/ltp>

⁸<https://github.com/lancopku/pkuseg-python>

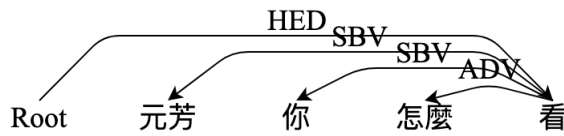


Figure 2.4: An example of dependency parsing in simplified Chinese using LTP.

tree the root node has a unique path to each lexical item.

An example of dependency parsing is shown in Figure 2.4. Root pointed to “看 (see)” and their relation has a label “HED”, which means “看 (see)” is the key item in this sentence. The relations between “看 (see)” and “元芳”, and, “看 (see)” and “你 (you)”, are labeled with “SBV”, which means their relations are subjects and verbs. As for the relation between “看 (see)” and “怎麼 (how)”, it is labeled with “ADV”, which means they have an adverbial relation.

There exist open-source tools for dependency parsing. In this thesis, dependency parsing from LTP is adopted and is used to help extract essential information from human’s utterances.

2.5 Recurrent Neural Network

A standard class of artificial neural networks is called Recurrent Neural Network (RNN), which inputs are temporal sequences and information of contexts is maintained by their internal states. RNN’s working principle is shown in Figure 2.5. The internal states, *i.e.* h_t in RNN is generated by both input elements, *i.e.* x_t , and previous internal states, *i.e.* h_{t-1} , through variable matrix W_x , W_h and an bias b . Then internal states are

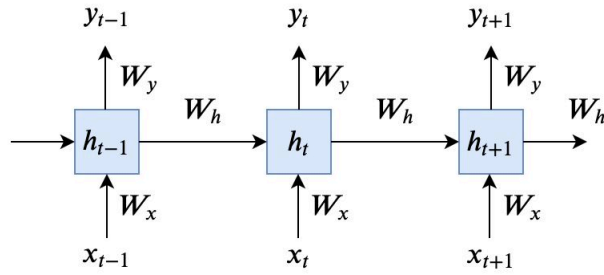


Figure 2.5: The working principle of unfolded RNN.

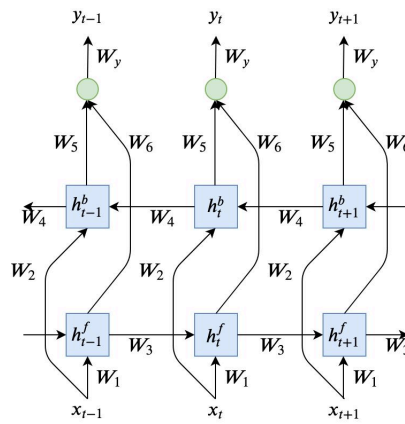


Figure 2.6: The working principle of unfolded BRNN.

transformed into outputs through W_y and an activation function $g(\cdot)$.

$$\begin{aligned}
 h_t &= W_h * h_{t-1} + W_x * x_t + b \\
 y_t &= g(W_y * h_t)
 \end{aligned}
 \tag{2.1}$$

An improvement on RNN model is Bidirectional RNN (BRNN) [36], which takes not only former contexts but also latter contexts into considerations. As shown in Figure 2.6, firstly, each inputs pass through a forward hidden layer and a backward hidden layer. Then the hidden states stored in both forward hidden layer and backward hidden layer are later concatenated (Denoted by green circle in Figure 2.6) and passed to the output layer to

generate outputs.



2.5.1 Long Short-Term Memory

Long short-term memory (LSTM) [37] is one of the RNN cells that is proposed to avoid the vanishing gradient problem as well as the gradient exploding problem of training an RNN model. The internal structure of an LSTM cell is shown in Figure 2.7. This structure can be explained using several gates. The first gate is called forget gate which decides whether to keep previous memory in the cell. Through W_f and a sigmoid function $\sigma(\cdot)$, the concatenation of h_{t-1} and x_t is transformed into f_t , obtaining a value between 0 and 1. If f_t is 0, then previous memory C_{t-1} will be completely forget. In contrast, if f_t is 1, then previous memory will be remained. The second gate is input gate, which is to decide what information from input x_t . Through i_t and \tilde{C}_t , memory C_t in the cell is updated. The last gate is output gate, which determines what to output. The output gate takes h_{t-1} and x_t as inputs and generate o_t , which is a value between 0 and 1. Then o_t decides what information from C_t will be transformed into h_t as the hidden state for this timestamp.

$$\begin{aligned}f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\h_t &= o_t * \tanh(C_t)\end{aligned}\tag{2.2}$$

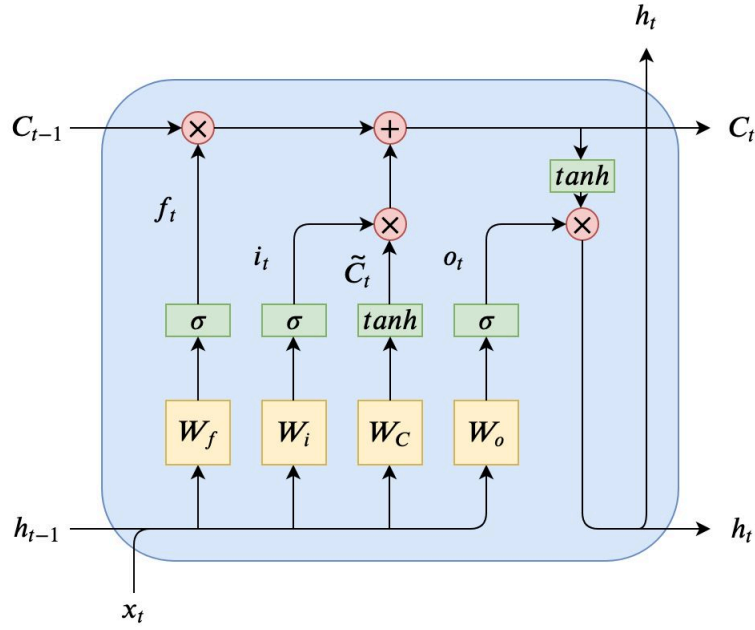


Figure 2.7: The internal structure of an LSTM cell.

2.5.2 Gated Recurrent Unit

Gated Recurrent Unit (GRU) [38] is an RNN cell adapted from LSTM which reduces the parameters needed as well as improves the speed of convergence while training. The internal structure of GRU is shown in Figure 2.8. Firstly, previous hidden state h_{t-1} and x_t pass through the reset gate and form r_t . They also pass through update gate and form z_t . r_t and z_t decide the portion of old memory and new input which together form current state h_t .

$$\begin{aligned}
 r_t &= \sigma(W_r * [h_{t-1}, x_t]) \\
 z_t &= \sigma(W_z * [h_{t-1}, x_t]) \\
 \tilde{h}_t &= \tanh(W_h * [r_t * h_{t-1}, x_t]) \\
 h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t
 \end{aligned} \tag{2.3}$$

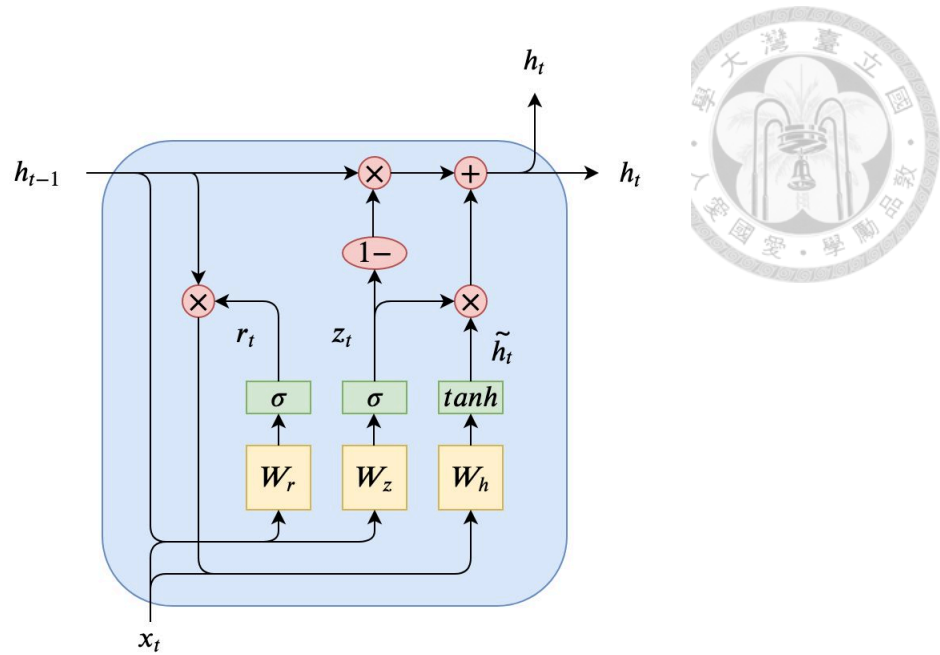


Figure 2.8: The internal structure of an GRU cell.

2.6 Sequence to Sequence

Sequence to Sequence (Seq2Seq) is a structure which can take sequences as both inputs and outputs. It consists of two main parts, the encoder and the decoder, which are both made up of RNN cells. As shown in Figure 2.9, the job of the encoder is to gather input sequences, *i.e.* x_1, x_2, x_3 and x_4 , and represent these inputs using a hidden vector (Indicated by an orange rectangle in Figure 2.9). Then the decoder utilizes the hidden vector together with a start input $\langle bos \rangle$ to generate outputs, *i.e.* y_1, y_2, y_3 and $\langle eos \rangle$. This architecture is commonly used to deal with NLP tasks like machine translation or neural response generation.

2.6.1 Seq2Seq with Attention

An improvement on Seq2Seq is the addition of attention mechanism [39] [40], which tackles the bottleneck of using a fixed-length vector to decode all outputs. Under the

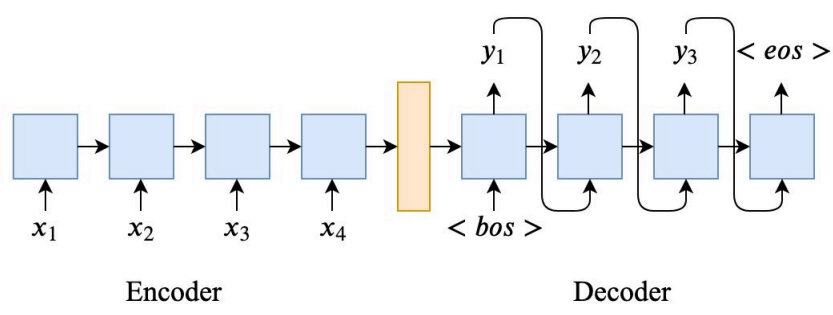


Figure 2.9: An illustration of Seq2Seq.

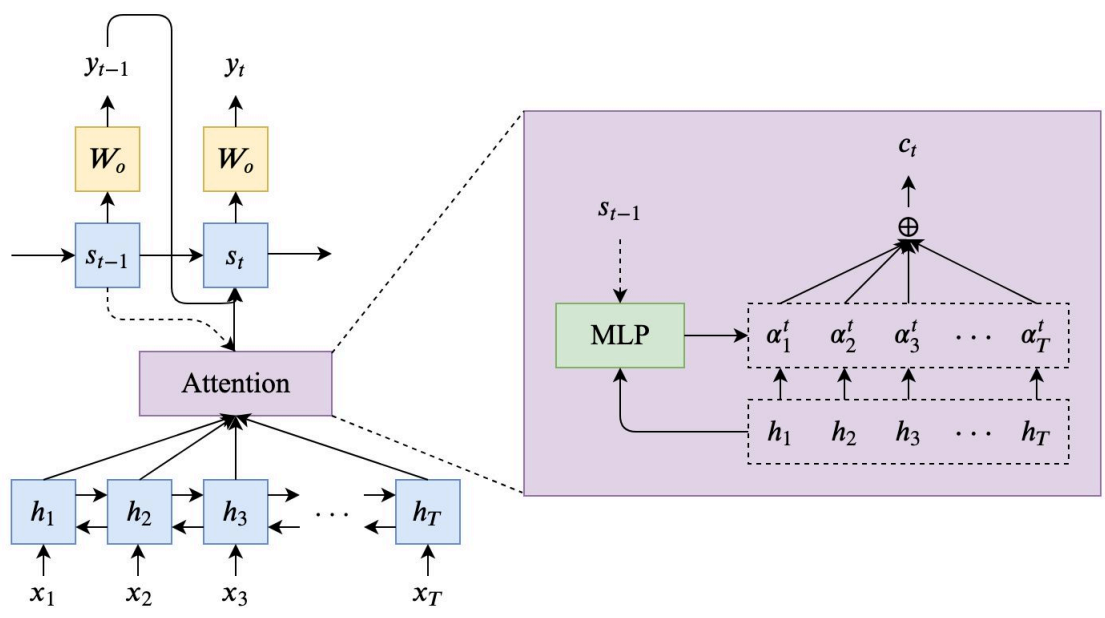


Figure 2.10: The working principle of an attention mechanism.

architecture of Seq2Seq with attention, the model is able to automatically search for parts in source sentences which are relevant to generated target words. The working principle of an attention mechanism is shown in Figure 2.10. The context vector c_t is the weighted sum of all encoded vectors h_i , and the weights α_i^t is a normalized similarity of s_{t-1} and h_i . The similarity is calculated using Multi-Layer Perceptron (MLP). Finally, s_t is generated by s_{t-1} , y_{t-1} and c_t through an RNN cell, which is denoted as $f(\cdot)$.

$$\begin{aligned}
 c_t &= \sum_{i=1}^T \alpha_i^t h_i \\
 \alpha_i^t &= \frac{\exp(e_i^t)}{\sum_{k=1}^T \exp(e_k^t)} \\
 e_i^t &= MLP(s_{t-1}, h_i) \\
 s_t &= f(s_{t-1}, y_{t-1}, c_t)
 \end{aligned} \tag{2.4}$$



Chapter 3

Methodology

3.1 System Overview

The proposed system can be divided into five major parts, as illustrated in Figure 3.1. Firstly, Natural Language Understanding (NLU) module will extract informative memory items from human utterances and provide sentiment analysis for the utterances. Next, the extracted information as well as the results of sentiment analysis will be updated to Global State Tracker (GST) module. In Dialogue Management (DM) module, GST module maintains information needed by other modules until the end of the dialogue. Meanwhile,

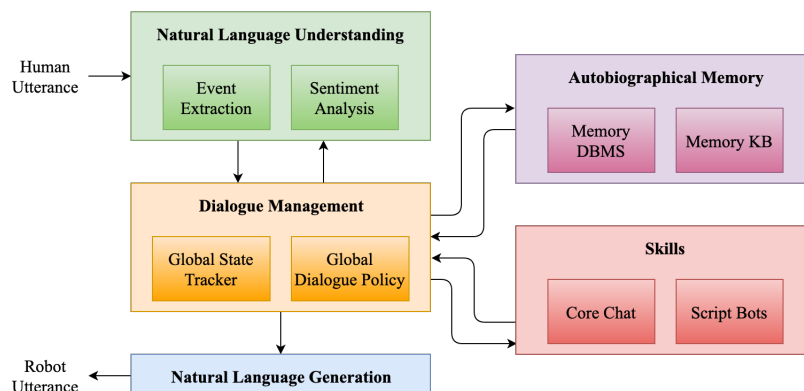
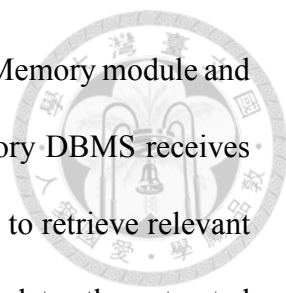


Figure 3.1: The architecture of overall system.



Global Dialogue Policy (GDP) module commands Autobiographical Memory module and Skills module, based on several information stored in GST. If Memory DBMS receives commands from GDP, it utilizes extracted information stored in GST to retrieve relevant memory items from Memory Knowledge Base (Memory KB) and updates the extracted information to Memory KB as well. The retrieved memory items are sent back to GST. As for Skills module, it consists of two sub modules, Core Chat module and Script Bots module. These sub modules obey the commands from GDP and generate robot's responses. The generated responses are also updated to GST. Finally, Natural Language Generation (NLG) module helps output robot's responses.

In the rest of this chapter, Autobiographical Memory module will be firstly introduced in Section 3.2, including Memory DBMS and Memory KB. Next the design of NLU is described in Section 3.3, including Event Extraction module, which helps to retrieve memory items from human utterances, and Sentiment Analysis module, which helps to predict human emotion. Then DM is introduced in Section 3.4, consisting of GST module and GDP module. After that, Skills module will be introduced in Section 3.5, including Core Chat module and Script Bots module. Finally, NLG module is described in Section 3.6, which helps to present robot's utterances to humans.

3.2 Autobiographical Memory

Autobiographical memory is a recollection of episodes from one's life, which can be viewed as an essential element of one's personal identity. In this thesis, autobiographical memory knowledge base introduced in Self-Memory System, which is a conceptual model of autobiographical memory as described in Section 1.3.1, is built into a computational model under the help of SQL (described in Section 2.1.1). Utilizing this computational

memory model, the proposed system is able to collect and manage human memory. Besides, robots are able to generate memory-related responses by utilizing memory items retrieved from Memory KB.



3.2.1 Memory Knowledge Base

After tracing the conceptual architecture of autobiographical memory knowledge base [1], a computational model of it is shown in Figure 3.2. The model is made of four layers, **Theme Layer**, **Lifetime Period Layer**, **General Event Layer**, and **Episodic Memory Layer**. Each layer is constructed by one or several SQL tables, and each table has specific primary keys as well as foreign keys to link to other tables. Contents and rules of these layers will be described in detail in this subsection.

• *Theme Layer*

Theme Layer separates human life into several subjects and is represented by `theme` table, as shown in Figure 3.3. In `theme` table, two attributes are given. One is `id`, which is the primary key, and the other one is `name` which stores the names of themes. In this work, **Theme Layer** contains only nine predefined themes, namely, “住所 (habitation)”, “旅遊 (travel)”, “家人 (family)”, “朋友 (friend)”, “伴侶 (soulmate)”, “讀書 (study)”, “工作 (work)”, “興趣 (hobby)”, and “其他 (others)”, as shown in Figure 3.2. One theme can not connect to other themes in **Theme Layer**, and it can only connect to specific lifetime periods in **Lifetime Period Layer**.

• *Lifetime Period Layer*

Lifetime Period Layer consists of lifetime periods and is represented by `lifetime` table, as shown in Figure 3.4. In `lifetime` table, the primary key consists of four at-

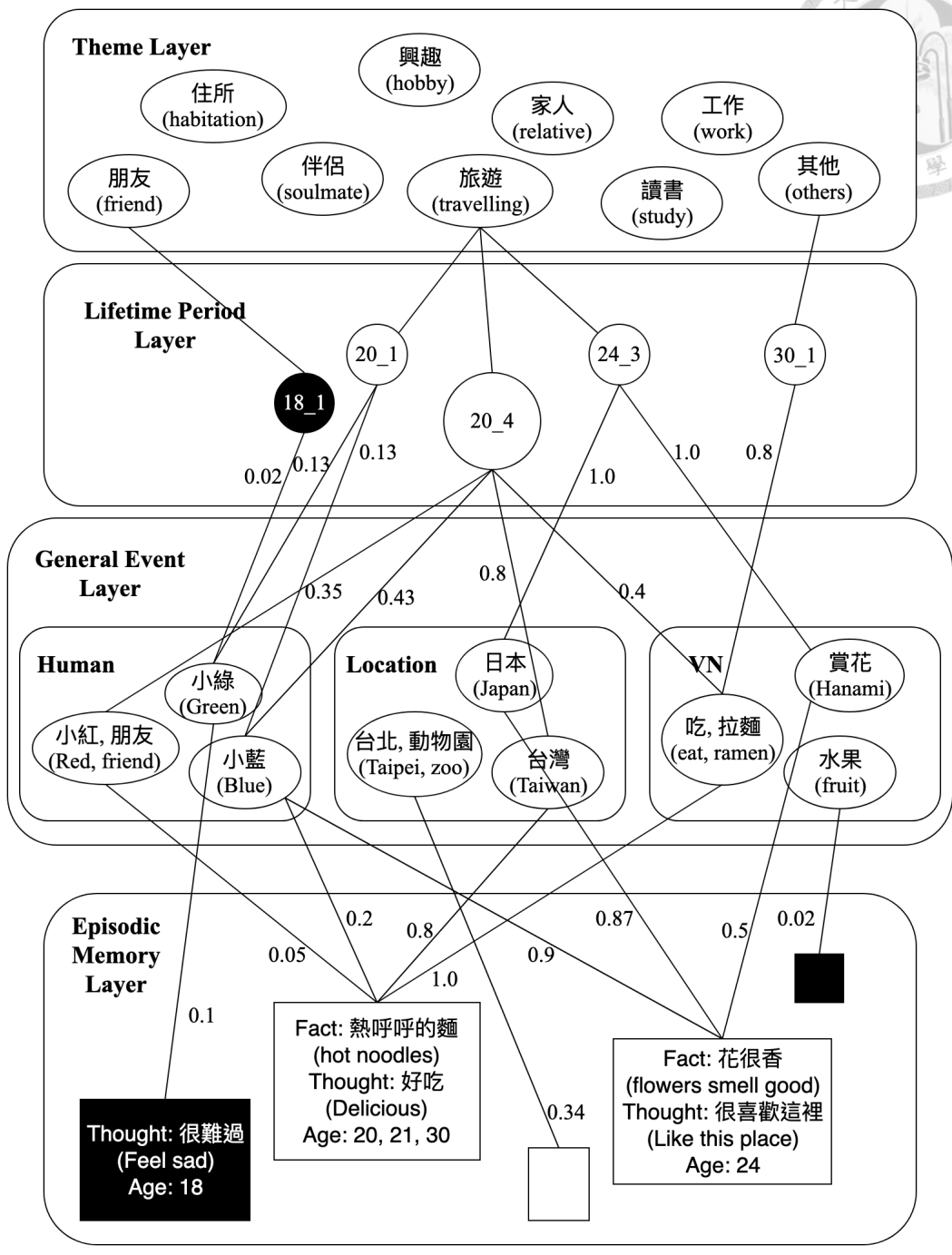


Figure 3.2: The architecture of Memory KB.

theme	
PK	id INT
	name VARCHAR

Figure 3.3: The relational schema of theme table.

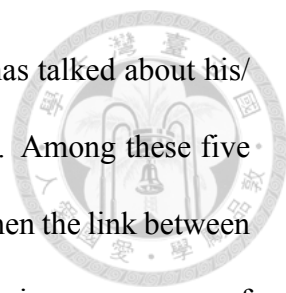


lifetime		
PK	theme_id	INT
PK	age	INT
PK	id	INT
PK	sub_id	INT
FK	human_id	INT
	count_h	INT
FK	location_id	INT
	count_l	INT
FK	vn_id	INT
	count_vn	INT
	sentiment	INT
	num_act	INT

Figure 3.4: The relational schema of lifetime table.

tributes, theme_id, age, id, and sub_id. theme_id represents the link between lifetime periods and themes in **Theme Layer**. age is used to arrange lifetime periods in the order of time and to distinguish similar items that happened in different years. The default value of age is set to -1 which is used to represent lifetime periods with unknown age. id, which can be viewed as the identity of items, identifying items that happened in the same year. Finally, sub_id is designed to enable one lifetime period to link to multiple items in **General Event Layer**. In Figure 3.2, nodes in **Lifetime Period Layer** are tagged with age and id. For example, a node tagged with “20_4” means that the lifetime period is the fourth item memorized in Memory KB in one’s twenty.

Links between lifetime periods and items in **General Event Layer** are represented by three attributes, human_id, location_id, vn_id and sentiment. Except for sentiment, each of them is a foreign key to human table, location table, and vn table respectively. Besides, each foreign key has a counter, count_h, count_l, and count_vn, which offers help in the procedure of retrieval of Memory DBMS. A counter together with num_act are used to compute the probability that an item will be mentioned



in a lifetime period. For example, as shown in Figure 3.2, a person has talked about his/her travelling experiences to Taiwan in his/her twenty for five times. Among these five times, if this person has mentioned “台灣” (Taiwan) for four times, then the link between this lifetime period and “台灣” (Taiwan) will be labelled with 0.8, since `num_act` of this lifetime period is five and `count_1` of “台灣” (Taiwan) under this lifetime period is four. It is reasonable that “台灣” (Taiwan) is more significant than “小藍” (Blue) with respect to this lifetime period, since the probability of occurrence of “台灣” (Taiwan) is higher than that of “小藍” (Blue).

`sentiment` is used to simply represent the evaluation of lifetime periods. If the sentiment is positive, `sentiment` will be set to 1, indicating by white circles in Figure 3.2. In contrast, if the sentiment is negative, `sentiment` will be set to -1, indicating by black circles. If the sentiment is neural, `sentiment` will be set to 0.

It is worth noticing that each lifetime period can only link to one specific theme in **Theme Layer** but can link to several items in **General Event Layer**. Besides, it's prohibited that a lifetime period directly connects to episodic memories in **Episodic Memory Layer**.

- **General Event Layer**

General Event Layer consists of three relation schemas, `human table`, `location table`, and `vn table` as shown in Figure 3.5, Figure 3.6 and Figure 3.7, which stores memory of humans, locations and activity & objects, respectively. Items in **General Event Layer** are linked by lifetime periods in **Lifetime Period Layer** and episodic memories in **Episodic Memory Layer**.

In `human table`, four attributes are given, `id`, `name`, `nickname` and `num_act`. `id`



human		
PK	id	INT
	name	VARCHAR
	nickname	VARCHAR
	num_act	INT

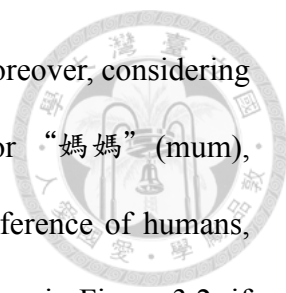
Figure 3.5: The relational schema of human table.

location		
PK	id	INT
	spe	VARCHAR
	gen	VARCHAR
	num_act	INT

Figure 3.6: The relational schema of location table.

vn		
PK	id	INT
	verb	VARCHAR
	noun	VARCHAR
	num_act	INT

Figure 3.7: The relational schema of vn table.



is the primary key of this table. `name` aims to store human names. Moreover, considering that humans may be called in other ways, such as “朋友” (dude) or “媽媽” (mum), `nickname` is introduced into this table. In order to know the preference of humans, `num_act` counts the times of humans being mentioned. For instance, in Figure 3.2, if “小藍” (Blue) is mentioned much more times than “小綠” (Green), then `num_act` of “小藍” (Blue) will be larger than that of “小綠” (Blue), indicated by a larger size of node.

In `location` table, four attributes are given as well, namely, `id`, `spe`, `gen` and `num_act`. `id` is the primary key. `spe` stores knowledge of special locations, such as “台灣” (Taiwan) or “台北” (Taipei), and `gen` stores knowledge of general locations, such as “動物園”(zoo) or “公園”(park). `num_act` learns the preference of locations. The larger `num_act` an item obtains, the higher probability that this location is much more familiar to a human. In Figure 3.2, for example, “台灣” (Taiwan) has larger `num_act` than “日本” (Japan), indicated by a larger size of node.

In `vn` table, similar to `human` table and `location` table, it contains four attributes, `id`, `verb`, `noun` and `num_act`. `id` is the primary key of this table. `verb` stores verbs, such as “吃”(eat) and “賞花”(hanami), while `noun` stores nouns, such as “拉麵”(ramen) and “水果”(fruit). `num_act`, similar to that of `human` table and that of `location` table, stores human preferences of activities and objects. Illustrated in Figure 3.2, the human prefers “水果”(fruit) to “賞花”(hanami).

Each item in **General Event Layer** can connect to several lifetime periods in **Lifetime Period Layer** and several episodic memories in **Episodic Memory Layer**. As for the relations between items, they can be inferred using **Lifetime Period Layer** and **Episodic Memory Layer**.



episodic		
PK	id	INT
PK	sub_id	INT
FK	human_id	INT
	count_h	INT
FK	location_id	INT
	count_l	INT
FK	vn_id	INT
	count_vn	INT
	thought	VARCHAR
	count_t	INT
	fact	VARCHAR
	count_f	INT
	sentiment	INT
	age	INT
	num_act	INT


Figure 3.8: The relational schema of `episodic` table.

• *Episodic Memory Layer*

Episodic Memory Layer aims to collect episodic memories into `episodic` table, as shown in Figure 3.8. Since episodic memories are much more specific than semantic knowledge stored in **Theme Layer**, **Lifetime Period Layer** and **General Event Layer**, the table requires more attributes to maintain abundant of information.

The primary key of this table is the combination of `id` and `sub_id`. `id` is the identity of episodic memories, and `sub_id` helps to connect one item with several general events in **General Event Layer**.

Descriptions of episodic memories are separated into several attributes: `human_id`, `location_id`, `vn_id`, `thought`, `fact`, `sentiment` and `age`. Among these attributes, `human_id`, `location_id` and `vn_id` are similar to those of `lifetime` table, which are foreign keys and link to tables in **General Event Layer**. `thought` aims to store some sighs or sentences full of sentiment like “好吃” (Delicious) and “很喜歡



這裡” (Like this place), and `fact` is designed to store things with modifiers like “熱呼呼的麵” (hot noodles) and “花很香” (flowers smell good). It is worth noticing that compared to `human`, `location` and `vn`, `thought` and `fact` are more unique with respect to each episodic memory, so that they are not stored using tables in `General Event Layer`. `sentiment` represents human sentiment using 1, 0 and -1. `age` not only stores temporal knowledge but also helps to map episodic memories to specific lifetime periods in **Lifetime Period Layer**. These descriptive attributes, except for `sentiment` and `age`, have counters to compute the times of items being mentioned with respect to each episodic memories.

As illustrated in Figure 3.2, an episodic memory links to “小紅, 朋友” (Red, friend) and “小藍” (Blue) in `human table`, “台灣” (Taiwan) in `location`, and “吃拉麵” (eat, ramen) in `vn`. Each link has a probability which indicates the importance of linked items to this episodic memory. And this episodic memory stores “熱呼呼的麵” (hot noodles) into `fact` and “好吃” (Delicious) into `thought`. The `age` stored in this episodic memory is 20, 21 and 30, and its `sentiment` is positive, indicated by a white block.

3.2.2 Memory Database Management System

Database management system (DBMS) uses DML to manipulate databases and control the accessibility to databases as introduced in 2.1.1. In this thesis, Memory DBMS aims to deal with the retrieval of memory items from Memory KB and the storage of new memory items into Memory KB. Therefore, instead of describing basic manipulations (*i.e.*, insert, update and delete), the method of retrieving memory items from Memory KB and storing new memory items into Memory KB is fully described.

The overall procedure of retrieval and storage of memory items is shown in Algo-

rithm 1. The inputs can be divided into five categories: Memory KB, *Store* command, *Threshold*, $GST_{eItemName}$ and $GST_{rItemName}$. *Store* command is used to decide whether to update old items or store new items into Memory KB. *Threshold* is used for election of episodic memories store in *episodic* table. $GST_{eItemName}$ indicates items extracted from human utterances and stored in GST, including GST_{eTheme} , GST_{eHuman} , $GST_{eLocation}$, GST_{eVN} , GST_{eAge} , GST_{eFact} , $GST_{eThought}$ and GST_{eSenti} . $GST_{rItemName}$ indicates buffers from GST which is to store retrieved items from Memory KB.

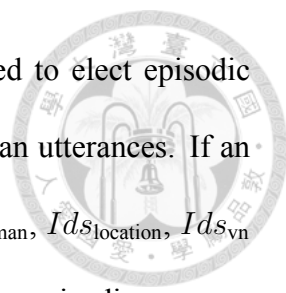
Algorithm 1 Memory DBMS algorithm

```

1: procedure MemoryDBMS(
   MemoryKB, Store, Threshold,
    $GST_{eTheme}$ ,  $GST_{eHuman}$ ,  $GST_{eLocation}$ ,  $GST_{eVN}$ ,
    $GST_{eAge}$ ,  $GST_{eFact}$ ,  $GST_{eThought}$ ,  $GST_{eSenti}$ ,
    $GST_{rTheme}$ ,  $GST_{rHuman}$ ,  $GST_{rLocation}$ ,  $GST_{rVN}$ ,  $GST_{rFact}$ ,  $GST_{rThought}$ )
2:    $Ids_{theme} \leftarrow \text{GetIds}(\text{MemoryKB}, GST_{eTheme})$ 
3:    $Ids_{human} \leftarrow \text{GetIds}(\text{MemoryKB}, GST_{eHuman}, \textit{Store})$ 
4:    $Ids_{location} \leftarrow \text{GetIds}(\text{MemoryKB}, GST_{eLocation}, \textit{Store})$ 
5:    $Ids_{VN} \leftarrow \text{GetIds}(\text{MemoryKB}, GST_{eVN}, \textit{Store})$ 
6:    $Ids_{EM} \leftarrow \text{VoteIds}(\text{MemoryKB}, \textit{Threshold}, Ids_{human}, Ids_{location}, Ids_{vn}, GST_{eAge})$ 
7:    $R \leftarrow \text{EMRetrieve}(\text{MemoryKB}, Ids_{EM})$ 
8:    $\text{MemoryKB} \leftarrow \text{MemoryKB} \cup \text{EMStore}(\text{MemoryKB},$ 
    $Ids_{EM}, Ids_{human}, Ids_{location}, Ids_{VN}, GST_{eAge}, GST_{eFact}, GST_{eThought}, GST_{eSenti}, \textit{Store})$ 
9:    $Ids_{LP} \leftarrow \text{VoteIds}(\text{MemoryKB}, \textit{Threshold},$ 
    $Id_{theme}, Ids_{human}, Ids_{location}, Ids_{vn}, GST_{eAge})$ 
10:   $R \leftarrow R \cup \text{LPRetrieve}(\text{MemoryKB}, Ids_{LP})$ 
11:   $\text{MemoryKB} \leftarrow \text{MemoryKB} \cup \text{LPStore}(\text{MemoryKB},$ 
    $Ids_{LP}, Ids_{human}, Ids_{location}, Ids_{VN}, GST_{eAge}, GST_{eSenti}, \textit{Store})$ 
12:   $R \leftarrow R \cup \text{GSTRetrieve}(\mathit{GST}_{eTheme}, \mathit{GST}_{eHuman}, \mathit{GST}_{eLocation}, \mathit{GST}_{eVN}, \mathit{GST}_{eFact}, \mathit{GST}_{eThought})$ 
13:   $\mathit{GST}_{rTheme}, \mathit{GST}_{rHuman}, \mathit{GST}_{rLocation},$ 
    $\mathit{GST}_{rVN}, \mathit{GST}_{rFact}, \mathit{GST}_{rThought} \leftarrow \text{GSTUpdate}(R)$ 
14:   $R \leftarrow \text{RClear}(R)$ 
15:  return  $\mathit{GST}_{rTheme}, \mathit{GST}_{rHuman}, \mathit{GST}_{rLocation}, \mathit{GST}_{rVN}, \mathit{GST}_{rFact}, \mathit{GST}_{rThought}$ 
16: end procedure

```

Firstly, in line 2-5, $Ids_{ItemName}$ of GST_{eTheme} , GST_{eHuman} , $GST_{eLocation}$ and GST_{eVN} are retrieved from theme table, human table, location table and vn table. If *Store* command is *True*, then these items will be updated and stored into Memory KB.



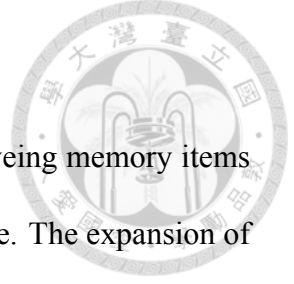
Then, in line 6, Ids_{human} , Ids_{location} , Ids_{vn} and GST_{eAge} are used to elect episodic memories stored in `episodic` table which may be relevant to human utterances. If an episodic memory in `episodic` table consists one element of Ids_{human} , Ids_{location} , Ids_{vn} and GST_{eAge} , the episodic memory gains one point. If the score of an episodic memory exceeds *Threshold*, its `id` is elected. Ids_{EM} is the collection of `id` of elected episodic memories sorted by their scores.

In line 7, Ids_{EM} are used to retrieve relevant memory items from `episodic` table, including `human`, `location`, `vn`, `thought` and `fact`, and the retrieved relevant memory items are temporarily stored in *R*. In line 8, if *Store* command is *True*, Ids_{ItemName} and the rest of $GST_{\text{eItemName}}$ will be updated and stored into `episodic` table.

In line 9-11 the same procedure in line 6-8 is used to retrieve relevant memory items from `lifetime` table and update $GST_{\text{eItemName}}$ into `lifetime` table.

After that, in line 12, $GST_{\text{eItemName}}$ are also updated to *R* and are viewed as a part of retrieved items. It is used to prevent the situation that nothing can be retrieved from Memory KB. During storytelling, if nothing can be retrieved from Memory KB, humans may confuse about whether robots are listening to them, since they can not receive any responses from robots. Therefore, $GST_{\text{eItemName}}$ is used to let people know that robots are kept listening and are trying to understand their utterances.

In line 13, retrieved items are updated to $GST_{\text{rItemName}}$, including GST_{rTheme} , GST_{rHuman} , $GST_{\text{rLocation}}$, GST_{rVN} , GST_{rFact} and GST_{rThought} . And in line 14, *R* is reset to an empty set. Finally, in line 15, GST_{rTheme} , GST_{rHuman} , $GST_{\text{rLocation}}$, GST_{rVN} , GST_{rFact} and GST_{rThought} are returned.



• **Retrieval and Storage of Episodic Memory Layer**

In Algorithm 1, line 7 and line 8 indicate the procedure of retrieving memory items from `episodic` table and updating new items into `episodic` table. The expansion of line 7 and line 8 in Algorithm 1 is shown in Algorithm 2 and Algorithm 3.

Algorithm 2 EMRetrieve Algorithm

```

1: procedure EMRetrieve(MemoryKB,  $Ids_{EM}$ )
2:   if  $Ids_{EM} \neq \emptyset$  then
3:      $OldItems \leftarrow$  RetrieveOldItems(MemoryKB,  $Ids_{EM}$ )
4:   end if
5:   return  $OldItems$ 
6: end procedure

```

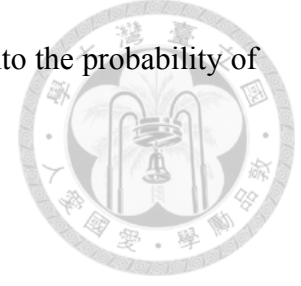
In Algorithm 2, if Ids_{EM} , which is the collection of `id` of episodic memories from `episodic` table, is not empty, items of these episodic memories are retrieved. Instead of equally retrieving all items, Bayes' theorem is adopted here to compute the probability of each item being retrieved, so as to increase the accuracy of retrieval of Memory KB.

Given a set of episodic memories, $EP = \{ep_j\}_{j=1}^M$, assume that episodic memories in EP are independent of each other, and a set of items of EP , $ITEM = \{item_i\}_{i=1}^N$, then the probability of observing a specific item $item_i$ is

$$\begin{aligned}
 p(item_i) &= \alpha_i \sum_{j=1}^M p(item_i|ep_j)p(ep_j) \\
 p(ep_j) &= \frac{n_j}{\sum_{k=1}^M n_k} \\
 p(item_i|ep_j) &= \frac{n_{ij}}{n_j}
 \end{aligned} \tag{3.1}$$

where $p(ep_j)$ is the probability of observing ep_j from EP , computed using n_j , which is `num_act` of ep_j . And $p(item_i|ep_j)$ is the probability of observing $item_i$ conditioned ep_j , computed through n_{ij} and n_j . n_{ij} indicates the count of $item_i$ of ep_j , including `count_h`, `count_l`, `count_vn`, `count_f` and `count_t`. α_i is a bias computed using scores of

episodic memories in order to inherit scores of episodic memories into the probability of items.



$$\alpha_i = \frac{\text{score}(\text{item}_i)}{\sum_{l=1}^N \text{score}(\text{item}_l)} \quad (3.2)$$

$$\text{score}(\text{item}_i) = \sum_{\{j:\text{item}_i \in \text{ep}_j\}} \text{score}_{\text{ep}_j}$$

Algorithm 3 EMStore Algorithm

```

1: procedure EMStore(
   MemoryKB,  $Ids_{EM}$ ,  $Ids_{human}$ ,  $Ids_{location}$ ,  $Ids_{vn}$ ,
    $GST_{eAge}$ ,  $GST_{eFact}$ ,  $GST_{eThought}$ ,  $GST_{eSenti}$ ,  $Store$ )
2:   if  $Store = True$  then
3:     if  $Ids_{EM} \neq \emptyset$  then
4:        $Id_{EM} \leftarrow \text{GetFirstElement}(Ids_{EM})$ 
5:       MemoryKB  $\leftarrow$  MemoryKB  $\cup$  UpdateSentiment( $Id_{EM}$ ,  $GST_{eSenti}$ )
6:       MemoryKB  $\leftarrow$  MemoryKB  $\cup$  UpdateNumAct( $Id_{EM}$ )
7:        $NewIds_{human} \leftarrow \text{FilterOldItems}(Id_{EM}, Ids_{human})$ 
8:        $NewIds_{location} \leftarrow \text{FilterOldItems}(Id_{EM}, Ids_{location})$ 
9:        $NewIds_{vn} \leftarrow \text{FilterOldItems}(Id_{EM}, Ids_{vn})$ 
10:       $NewItems_{fact} \leftarrow \text{FilterOldItems}(Id_{EM}, Items_{fact})$ 
11:       $NewItems_{thought} \leftarrow \text{FilterOldItems}(Id_{EM}, Items_{thought})$ 
12:       $NumAct \leftarrow \text{GetNumAct}(Id_{EM})$ 
13:      MemoryKB  $\leftarrow$  MemoryKB  $\cup$  StoreNewItems(
         $Id_{EM}$ ,  $NewIds_{human}$ ,  $GST_{eAge}$ ,  $GST_{eSenti}$ ,  $NumAct$ )
14:      MemoryKB  $\leftarrow$  MemoryKB  $\cup$  StoreNewItems(
         $Id_{EM}$ ,  $NewIds_{location}$ ,  $GST_{eAge}$ ,  $GST_{eSenti}$ ,  $NumAct$ )
15:      MemoryKB  $\leftarrow$  MemoryKB  $\cup$  StoreNewItems(
         $Id_{EM}$ ,  $NewIds_{vn}$ ,  $GST_{eAge}$ ,  $GST_{eSenti}$ ,  $NumAct$ )
16:      MemoryKB  $\leftarrow$  MemoryKB  $\cup$  StoreNewItems(
         $Id_{EM}$ ,  $NewItems_{fact}$ ,  $GST_{eAge}$ ,  $GST_{eSenti}$ ,  $NumAct$ )
17:      MemoryKB  $\leftarrow$  MemoryKB  $\cup$  StoreNewItems(
         $Id_{EM}$ ,  $NewItems_{thought}$ ,  $GST_{eAge}$ ,  $GST_{eSenti}$ ,  $NumAct$ )
18:     else
19:        $Id_{EM} \leftarrow \text{GetMaxEMId}() + 1$ 
20:       MemoryKB  $\leftarrow$  MemoryKB  $\cup$  StoreNewEM(
         $Id_{EM}$ ,  $Ids_{human}$ ,  $Ids_{location}$ ,  $Ids_{vn}$ ,  $GST_{eAge}$ ,  $GST_{eFact}$ ,  $GST_{eThought}$ ,  $GST_{eSenti}$ )
21:     end if
22:   end if
23:   return MemoryKB
24: end procedure

```

In Algorithm 3, if $Store$ command is $True$, all input memory items will be updated



to MemoryKB, including a set of `id` from `human` table, a set of `id` from `location` table, a set of `id` from `vn` table, an extracted age, extracted facts, extracted thoughts and detected sentiments stored in GST.

If Id_{EM} is not empty, in line 4, the first element of the collection of `id` of elected episodic memories are chosen, which gains the highest score among all `id`, and is viewed as `id` for all given items, that is, all given items is viewed as items of this specific episodic memory. In line 5-6, `sentiment` and `num_act` of the chosen episodic memory is updated. From line 7-11, all given items are passed through a filter to filter out existing items of the chosen episodic memory in Memory KB and update their counts (*i.e.*, `count_h`, `count_l`, `count_vn`, `count_f` and `count_t`) as well. From line 13-17, new items of the chosen episodic memory are stored into the Memory KB.

As for the situation that Id_{EM} is an empty set, in line 19-20, a new `id` is generated and all given items will be directly updated to this new episodic memory.

• *Retrieval and Storage of Lifetime Period Layer*

The procedure of retrieving old items from `lifetime` table and storing new items into it is similar to those of `episodic` table. The expansion of line 10 and line 11 in Algorithm 1 is shown in Algorithm 2 and Algorithm 3.

In Algorithm 1, line 7 and line 8 indicate the procedure of retrieving memory items from `episodic` table and updating new items into `episodic` table. The expansion of line 7 and line 8 in Algorithm 1 is shown in Algorithm 4 and Algorithm 5.

In Algorithm 4, if Ids_{LP} , which is the collection of tuples made up of `theme_id`, age, `id` of lifetime periods from `lifetime` table, is not empty, items of these lifetime periods are retrieved. Instead of equally retrieving all items, Bayes' theorem is adopted



Algorithm 4 LPRetrieve Algorithm

```

1: procedure LPRetrieve(MemoryKB,  $Ids_{LP}$ )
2:   if  $Ids_{LP} \neq \emptyset$  then
3:      $OldItems \leftarrow$  RetrieveOldItems(MemoryKB,  $Ids_{LP}$ )
4:   end if
5:   return  $OldItems$ 
6: end procedure

```

here to compute the probability of each item being retrieved, so as to increase the accuracy of retrieval of Memory KB.

Given a set of lifetime periods, $LP = \{lp_j\}_{j=1}^M$, assume that lifetime periods in LP are independent of each other, and a set of items of LP , $ITEM = \{item_i\}_{i=1}^N$, then similar to Equation 3.1 and Equation 3.2 the probability of observing a specific item $item_i$ is

$$\begin{aligned}
 p(item_i) &= \alpha_i \sum_{j=1}^M p(item_i|lp_j)p(lp_j) \\
 p(lp_j) &= \frac{n_j}{\sum_{k=1}^M n_k} \\
 p(item_i|lp_j) &= \frac{n_{ij}}{n_j}
 \end{aligned} \tag{3.3}$$

where $p(lp_j)$ is the probability of observing lp_j from LP , computed using n_j , num_act of lp_j . And $p(item_i|lp_j)$ is the probability of observing $item_i$ conditioned lp_j , computed using n_{ij} and n_j . n_{ij} indicates the count of $item_i$ of lp_j , including count_h, count_l and count_vn. α_i is a bias computed using scores of lifetime periods in order to merge scores of lifetime periods into the probability of items.

$$\begin{aligned}
 \alpha_i &= \frac{score(item_i)}{\sum_{l=1}^M score(item_l)} \\
 score(item_i) &= \sum_{\{j:item_i \in lp_j\}} score_{lp_j}
 \end{aligned} \tag{3.4}$$

In Algorithm 5, if *Store* command is *True*, all input memory items will be updated to MemoryKB, including a set of id from human table, a set of id from location table,



Algorithm 5 LPStore Algorithm

```
1: procedure LPStore(  
   MemoryKB,  $Ids_{EM}$ ,  $Ids_{human}$ ,  $Ids_{location}$ ,  $Ids_{vn}$ ,  
    $GST_{eAge}$ ,  $GST_{eSenti}$ ,  $Store$ )  
2:   if  $Store = True$  then  
3:     if  $Ids_{LP} \neq \emptyset$  then  
4:        $Id_{LP} \leftarrow \text{GetFirstElement}(Ids_{LP})$   
5:       MemoryKB  $\leftarrow$  MemoryKB  $\cup$  UpdateSentiment( $Id_{LP}$ ,  $GST_{eSenti}$ )  
6:       MemoryKB  $\leftarrow$  MemoryKB  $\cup$  UpdateNumAct( $Id_{LP}$ )  
7:        $NewIds_{human} \leftarrow \text{FilterOldItems}(Id_{LP}, Ids_{human})$   
8:        $NewIds_{location} \leftarrow \text{FilterOldItems}(Id_{LP}, Ids_{location})$   
9:        $NewIds_{vn} \leftarrow \text{FilterOldItems}(Id_{LP}, Ids_{vn})$   
10:       $NumAct \leftarrow \text{GetNumAct}(Id_{LP})$   
11:      MemoryKB  $\leftarrow$  MemoryKB  $\cup$  StoreNewItems(  
         $Id_{LP}$ ,  $NewIds_{human}$ ,  $GST_{eSenti}$ ,  $NumAct$ )  
12:      MemoryKB  $\leftarrow$  MemoryKB  $\cup$  StoreNewItems(  
         $Id_{LP}$ ,  $NewIds_{location}$ ,  $GST_{eSenti}$ ,  $NumAct$ )  
13:      MemoryKB  $\leftarrow$  MemoryKB  $\cup$  StoreNewItems(  
         $Id_{LP}$ ,  $NewIds_{vn}$ ,  $GST_{eSenti}$ ,  $NumAct$ )  
14:     else  
15:        $Id_{LP} \leftarrow \text{GetMaxLPid}() + 1$   
16:       MemoryKB  $\leftarrow$  MemoryKB  $\cup$  StoreNewLP(  
         $Id_{LP}$ ,  $Ids_{human}$ ,  $Ids_{location}$ ,  $Ids_{vn}$ ,  $GST_{eSenti}$ )  
17:     end if  
18:   end if  
19:   return MemoryKB  
20: end procedure
```

a set of `id` from `vn` table, and detected sentiments stored in GST.

If Id_{LM} is not empty, in line 4, the first element of the collection of elected lifetime periods are chosen, which gains the highest score, and is viewed as the lifetime period for all given items. In line 5-6, `sentiment` and `num_act` of the chosen lifetime period is updated. From line 7-9, all given items are passed through a filter to filter out existing items of the chosen lifetime period in Memory KB and update their counts (*i.e.*, `count_h`, `count_l` and `count_vn`) as well. From line 11-13, new items of the chosen episodic memory are stored into the Memory KB.

As for the situation that Id_{LP} is an empty set, in line 15-16, a new lifetime period is generated and all given items will be directly updated to this new lifetime period.

3.3 Natural Language Understanding

NLU module is one of the most essential components to a dialogue system. It helps a system to parse human utterances and its parsing results may have large influence on the performance of the overall dialogue system [41]. Classical NLU module performs three tasks: slot tagging, domain detection and intent determination.

However, in this thesis, instead of developing these techniques, the main task to solve is event extraction, which is to extract memory events from human's utterances in order to meet the need of Memory KB. Another task is to perform sentiment analysis in order to detect human feelings. Two modules are introduced in this section. One is Event Extraction module and the other one is Sentiment Analysis module.

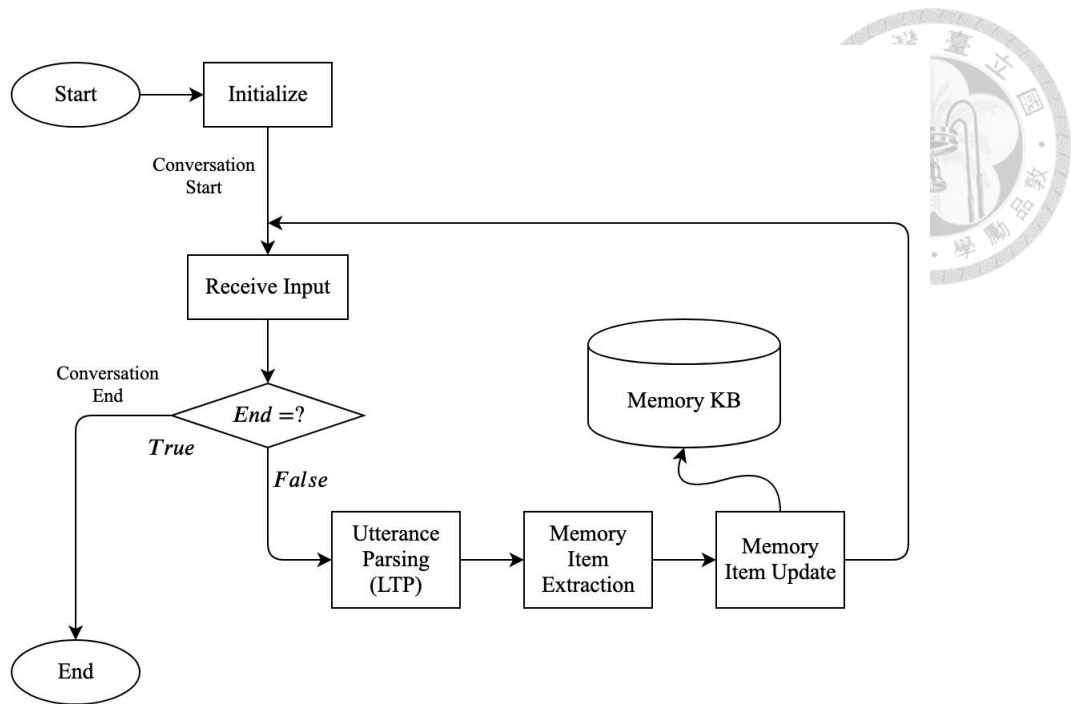


Figure 3.9: The workflow of Event Extraction Module.

3.3.1 Event Extraction

Event extraction is a category under information extraction which is to identify events such as who, what, when, where, why, how and even to detect the relationship between these events such as who did what to whom. It is considered as the hardest one among all information extraction tasks [42]. In this thesis, the proposed Event Extraction module is based on pattern-matching methods as other event extraction works in Chinese [43] [44]. The workflow is shown in Figure 3.9.

Initially, several parameters are given to initialize Entity Extraction module, including the age of human, the year and thresholds for event extraction. Next, the module receives inputs. An input consists of four components: human utterances, detected sentiments from Sentiment Analysis module, and Global State Tracker. After receiving inputs, it checks human's intent of chatting. The module won't do anything unless human intents are willing to chat. If humans are willing to chat, then input utterances will be passed

through three functions, Utterance Parsing, Memory Item Extraction and Memory Item Update. The three functions will be described in detail.



• **Utterance Parsing**

Instead of developing new tools for parsing, tools from LTP¹, a well-known Chinese language platform, is adopted. Human utterances is firstly segmented into words. Then POS tagging from LTP is applied, which example is shown in Figure 2.3. Later on, dependency parsing, introduced in Section 2.4, is applied. And through Algorithm 6, dependency parsing gives scores to each word.

Algorithm 6 Dependency Parsing Scoring Algorithm

```

1: procedure DependencyParsingScoring(Words, POSs)
2:   Heads  $\leftarrow$  0
3:   Rank  $\leftarrow$  1
4:   ScoreList  $\leftarrow$  []
5:   Pars  $\leftarrow$  DependencyParsing(Words, POSs)
6:   while True do
7:     Heads  $\leftarrow$  FindHeads(Heads, Pars)
8:     if Heads  $\neq$   $\emptyset$  then
9:       Scoresheads  $\leftarrow$  Reciprocal(Rank)
10:      Heads  $\leftarrow$  Heads
11:      Rank  $\leftarrow$  Rank + 1
12:      ScoreList  $\leftarrow$  UpdateToScoreList(ScoreList, Scoresheads)
13:     else
14:       break
15:     end if
16:   end while
17:   ScoreList  $\leftarrow$  Normalize(ScoreList)
18:   return ScoreList
19: end procedure

```

As shown in Algorithm 6, words and the results of POS tagging are the inputs. In line 2-4, some variables are initialized: *Heads* is initialized to 0 which means the root, *Rank* is initialized to 1 which means one step away from the root, and, *ScoreList* is initialized to an empty list. In line 5, words and the results of POS tagging are passed through dependency

¹<https://github.com/HIT-SCIR/ltp>

parsing from LTP. *Pars* stores the results of dependency parsing, each element in *Pars* consists of the position of its head and the relationship between them.

In line 6-16, scores of words are computed. Firstly, in line 7, words that directly link to heads are chosen and are viewed as new heads. For example, if a word directly link to the root, then this word will be chosen and it will be denoted as the new head. Then, if chosen words exist, the reciprocal of *Rank* is assigned as scores of chosen words. It is reasonable that the closer a word to the root, the more important it is in a sentence. After that, in line 10-12, variable are updated.

Finally, before returning scores of words, in line 17, all the scores are normalized.

• *Memory Item Extraction*

Item Extraction is to extract memory items from human utterances. These items include theme, human, location, activity & object, fact, thought and age, with respect to human table, location table, vn table, fact and thought in episodic table, and age in lifetime table and episodic table. To accomplish this task, pattern-matching are adopted in this thesis. With the help of POS tagging from LTP, each word is tagged with a tag. For example, “你” (you) is tagged as “r” , which means pronoun. The tags of POS tagging from LTP is listed in Table 3.1, together with their explanations and examples.

Patterns for extracting memory items are designed based on tags of POS tagging. Each type of items is extracted using uni-gram patterns, bi-gram patterns, tri-gram patterns and other special patterns if needed.

To extract items of human, which may be stored into human table in Memor KB, the patterns are shown in Table 3.2. It mainly depends on the tag “nh” and a collected list of

²<http://ltp.ai/docs/appendix.html>



Table 3.1: Tags of POS tagging from LTP. (Adapted from ².)

Tag	Explanation	Example
a	adjective	美麗 (beautiful)
b	other noun-modifier	西式 (western)
c	conjunction	雖然 (although)
d	adverb	很 (very)
e	exclamation	唉 (Ugh)
g	morpheme	甥
h	prefix	阿
i	idiom	百花齊放
j	abbreviation	公檢法
k	suffix	率
m	number	一 (one)
n	general noun	蘋果 (apple)
nd	direction noun	右側 (right)
nh	person name	湯姆 (Tom)
ni	organization name	台積電 (TSMC)
nl	location name	城郊 (country)
ns	geographical name	台北 (Taipei)
nt	temporal noun	今天 (today)
nz	other proper name	諾貝爾獎 (Nobel Prize)
o	onomatopoeia	嘩啦
p	preposition	在 (at)
q	quantity	束 (bundle)
r	pronoun	我們 (we)
u	auxiliary	的 (of)
v	verb	跑 (run)
wp	punctuation	， (,)
ws	foreign words	CPU (中央處理器)
x	non-lexeme	荀



Table 3.2: Rules for extracting items of human.

Category	Pattern
Uni-gram	$word \rightarrow \text{name}$, if $word$ tagged with nh ; $word \rightarrow \text{nickname}$, if $word \in \text{HumanList}$;
Bi-gram	$word_1 + word_2 \rightarrow \text{name} + \text{nickname}$, if $word_1$ tagged with nh/nz , and $word_2 \in \text{HumanList}$; $word_1 + word_2 \rightarrow \text{nickname} + \text{name}$, if $word_1 \in \text{HumanList}$ and $word_2$ tagged with nh/nz ;
Special Case	$word_1 + word_2 \rightarrow \text{name} + \text{nickname}$, if only $word_1$ tagged with nh , and only $word_2 \in \text{HumanList}$, and 叫/叫做/姓名/名字/稱呼/稱 (call) exists;

human nouns. The collected list contains words like “媽媽” (mother), “朋友” (friend) or “家人” (family member), which are tagged with “n” but most of time are used when talking about someone else. The design of bi-gram patterns is because that in some cases, humans are called like “珍妮阿姨” (Aunt Jenny) or “同學小綠” (Classmate Green). As for the design of special case, humans are used to introduce someone using sentences like “我朋友叫小藍” (My friend is called Blue). Therefore, the pattern for the special case is to detect the existence of words like “朋友”, “叫” and “小藍”. Moreover, the number of words tagged with “nh” or existed in the collected list is limited to one, respectively, in order to avoid any ambiguity.

To extract items of location, which may be stored into `location` table in Memory KB, the patterns are shown in Table 3.3. It mainly depends on the tag “ns” and “nl”, where “ns” marks words like “台北” (Taipei) and “nl” marks words like “山上”. However, it’s far from enough using the two tags to extract expressions of locations, because most of the places are tagged with “n”, such as “公園” (park), “學校” (school), or “動物園” (zoo). In order to extract places tagged with “n”, we ask help from “在” (at), “來”



Table 3.3: Rules for extracting items of location.

Category	Pattern
Uni-gram	$word \rightarrow spe$, if $word$ tagged with ns; $word \rightarrow gen$, if $word$ tagged with nl;
Bi-gram	$word \rightarrow gen$, if 在/來/去 (at/come/go) + $word$ and $word$ tagged with n; $word \rightarrow spe$, if 在/來/去 (at/come/go) + $word$ and $word$ tagged with j/ni/nz; $word_1 + word_2 \rightarrow spe + gen$, if $word_1$ tagged with ns, and $word_2$ tagged with nl; $word_1 + word_2 \rightarrow gen + spe$, if $word_1$ tagged with n/nl, and $word_2$ tagged with ns; $word_1 + word_2 \rightarrow spe$, if $word_1$ and $word_2$ tagged with ns;
Tri-gram	$word_1 + word_3 \rightarrow spe + gen$, if $word_1$ tagged with ns, and $word_2$ tagged with u, and $word_3$ tagged with nl;



Table 3.4: Rules for extracting items of activity & object.

Category	Rules
Uni-gram	$word \rightarrow \text{verb}$, if $word$ tagged with v ; $word \rightarrow \text{noun}$, if $word$ tagged with $n/ni/nz/j/nt$;
Bi-gram	$word_1 + word_2 \rightarrow \text{verb} + \text{noun}$, if $word_1$ tagged with v and $word_2$ tagged with $n/ni/nz/j/nt$; $word_1 + word_2 \rightarrow \text{noun}$, if $word_1$ and $word_2$ tagged with $n/ni/nz/j/nt$; $word_1 + word_2 \rightarrow \text{verb}$, if $word_1$ and $word_2$ tagged with v ;
Tri-gram	$word_1 + word_2 + word_3 \rightarrow \text{verb} + \text{noun}$, if $word_1$ and $word_2$ tagged with v , and $word_3$ tagged with $n/ni/nz/j/nt$; $word_1 + word_2 + word_3 \rightarrow \text{verb} + \text{noun}$, if $word_1$ tagged with v , and $word_2$ and $word_3$ tagged with $n/ni/nz/j/nt$; $word_1 + word_2 + word_3 \rightarrow \text{noun}$, if $word_1, word_2$ and $word_3$ tagged with $n/ni/nz/j/nt$;

(come) and “去” (go), which are words that humans are used to add places after them, such as “在學校” (at school), “來學校” (come to school) and “去學校” (go to school). Moreover, in some cases, general places may be specified by cities or countries, like “台灣山上” (mountain of Taiwan), so that a bi-gram rule is designed to extract this kind of expressions. The tri-gram pattern takes “的” (of) into considerations, which is tagged with “u” , and is used to extract expressions like “台北的動物園” (Zoo of Taipei).

To extract items of activity & object, which may be stored into vn table in Memory KB, the patterns are shown in Table 3.4. The reason that using the same patterns to extract expressions of activities and those of objects is that it’s hard to clearly separate them using only tags. An activity can be tagged with only “v” or a combination of “v” and “n” . In Table 3.4, “v” are used to extract expressions of verbs whereas “n” / “ni” / “nz” / “j” / “nt”

are used to extract expressions of nouns. Since expressions of activities and objects may consist of multiple words, bi-gram patterns and tri-gram patterns are designed as well.

To extract items of fact, which may be stored into *episodic* table in Memory KB, the patterns are shown in Table 3.5. Since the design of fact aims to store special events, the patterns are expected to extract expressions of activities and objects with modifiers. Bi-gram patterns are able to extract expressions like “高山” (high mountain) and “快跑” (fast run). Tri-gram and Quad-gram patterns takes “u” into considerations. Words like “的”, “地”, and “得” usually appear between modifiers and verbs or nouns. Therefore, words tagged with “u” are allowed to exist in the middle of an expression, such as “漂亮的花” (beautiful flower) or “非常迅速地跑” (very fast run).

To extract items of age, which may be stored into both *lifetime* table and *episodic* table in Memory KB, the patterns are shown in Table 3.6. The expressions of age can be roughly divided into two categories. One is directly using “歲” (age), the other one is using “年” (year). The expressions containing “歲” (age) can be extracted using bi-gram patterns, whereas the expressions containing “年” (year) is much more complicated. In some cases, expressions containing “年” (year) can be extracted using uni-gram patterns like “一九九五年” (1995). However, some expressions containing “年” (year) are divided into multiple tags, which results in the design of bi-gram, tri-gram and quad-gram patterns. Moreover, rules to distinguish “民國” (Republic Era) and “公元” (A.D.) are also added. Sometimes when humans mention about their past experiences, they are used to use expressions like “在十年前” (ten years ago). Therefore, patterns are also designed for expressions containing “前” or “之前” (ago).

After extracting various items, a biased term frequency-inverse document frequency (tf-idf) is used to rank the extracted items, in order to extract memory items of a paragraph



Table 3.5: Rules for extracting items of fact.

Category	Rules
Bi-gram	$word_1 + words_2 \rightarrow \text{fact},$ if $word_1$ tagged with a^* and $word_2$ tagged with n^* ; $word_1 + word_2 \rightarrow \text{fact},$ if $word_1$ tagged with a^* and $word_2$ tagged with v ;
Tri-gram	$word_1 + word_2 + word_3 \rightarrow \text{fact},$ if $word_1$ tagged with a^* , $word_2$ tagged with a^*/u and $word_3$ tagged with n^* ; $word_1 + word_2 + word_3 \rightarrow \text{fact},$ if $word_1$ tagged with n^* , $word_2$ tagged with a^*/u and $word_3$ tagged with a^* ; $word_1 + word_2 + word_3 \rightarrow \text{fact},$ if $word_1$ tagged with a^* , $word_2$ tagged with a^*/u , and $word_3$ tagged with v ; $word_1 + word_2 + word_3 \rightarrow \text{fact},$ if $word_1$ tagged with v , $word_2$ tagged with a^*/u , and $word_3$ tagged with a^* ;
Quad-gram	$word_1 + word_2 + word_3 + word_4 \rightarrow \text{fact},$ if $word_1$ tagged with a^* , $word_2$ and $word_3$ tagged with a^*/u , and $word_4$ tagged with n^* ; $word_1 + word_2 + word_3 + word_4 \rightarrow \text{fact},$ if $word_1$ tagged with n^* , $word_2$ and $word_3$ tagged with a^*/u , and $word_4$ tagged with a^* ; $word_1 + word_2 + word_3 + word_4 \rightarrow \text{fact},$ if $word_1$ tagged with a^* , $word_2$ and $word_3$ tagged with a^*/u , and $word_4$ tagged with v ; $word_1 + word_2 + word_3 + word_4 \rightarrow \text{fact},$ if $word_1$ tagged with v , $word_2$ and $word_3$ tagged with a^*/u , and $word_4$ tagged with a^* ;
Notes	n^* denotes n/j/nh/ni/nl/ns/nt/nz; a^* denotes a/b/d/i/z;



Table 3.6: Rules for extracting items of age.

Category	Rules
Uni-gram	$word \rightarrow age$, if $word$ tagged with nt and 年 (year) in $word$; $word \rightarrow age$, if $word$ in <i>TimeList</i> ;
Bi-gram	$word_1 + word_2 \rightarrow age$, if $word_1$ tagged with m and $word_2$ is 歲 (age); $word_1 + word_2 \rightarrow age$, if $word_1$ tagged with m and $word_2$ is 年 (year);
Tri-gram	$word_1 + word_2 + word_3 \rightarrow age$, if $word_1$ and $word_2$ tagged with m and $word_3$ is 年 (year); $word_1 + word_2 + word_3 \rightarrow age$, if $word_1$ tagged with m and $word_2$ is 年 (year), and $word_3$ is 前/之前 (before);
Quad-gram	$word_1 + word_2 + word_3 + word_4 \rightarrow age$, if $word_1, word_2$ and $word_3$ tagged with m and $word_4$ is 年 (year);

instead of a sentence. Given a set of sentences $S = \{s_j\}_{j=1}^M$ and sets of words $W_j = \{w_{ij}\}_{i=1}^{N_j}$ for each sentence, the biased tf-idf score of w_{ij} is computed as

$$\begin{aligned}
 tfidf_{bias}(w_{ij}) &= \beta_{ij} * tfidf(w_{ij}) \\
 tfidf(w_{ij}) &= tf(w_{ij}) * idf(w_{ij}) \\
 tf(w_{ij}) &= \sum_{k=1}^M \frac{n_{ik}}{\sum_{l=1}^{N_k} n_{lk}} \\
 idf(w_{ij}) &= \log \frac{M}{|\{j : w_{ij} \in s_j\}|} \\
 \beta_{ij} &= \sum_{j:w_{ij} \in s_j} \frac{N_j}{\sum_{k=1}^M N_k}
 \end{aligned} \tag{3.5}$$

where n_{ij} is the count of w_i in s_j .

The reason of adding bias to tfidf score is that long utterances should be more informative than short utterances. Therefore, words which belong to long utterances should

gain higher scores than words which belong to short utterances.



3.3.2 Sentiment Analysis

Sentiment Analysis is to detect human sentiment and the results of it are both stored into Memory KB. Besides, if a sequence of negative sentiment is detected, the system will generate suggestions to change to a new topic or to tell a new story. This helps not only prevent humans from reminding of and talking about sad experiences but also prevent robots keep making mistakes which makes human unhappy [5].

In this thesis, CopeOpi [45] is directly adopted to perform analysis on sentiment, which is able to perform sentiment analysis on texts in both Traditional Chinese and Simple Chinses.

3.4 Dialogue Management

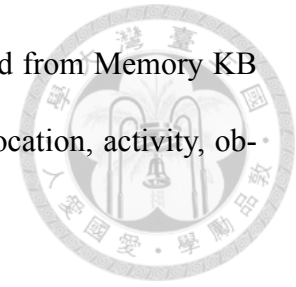
DM module controls the whole process in this system. It consists of GST module and GDP module. GST module maintains necessary information for other modules and GDP module generates commands to other modules.

3.4.1 Global State Tracker

GST module preserves information for other modules. The information can be categorized into several parts:

- **Storage for NLU**: a storage to maintain items from human's utterances extracted by NLU module, including items of theme, human, location, activity, object, fact, age, thought and sentiment.

- ***Storage for Memory KB***: a storage to maintain items retrieved from Memory KB through Memory DBMS, including items of theme, human, location, activity, object, fact and thought.
- ***Storage for GDP***: a storage to maintain necessary information needed by GDP module, including human input utterances, the state of human identification, and a global state which affects the workflow of the overall system.
- ***Storage for Skills***: a storage to maintain necessary information needed by Skills module as well as generated responses.



3.4.2 Global Dialogue Policy

GDP module controls the workflow of the overall system and give commands to other modules. The commands given by GDP module are in a hierarchical structure, denoted as the top-level commands and the low-level commands.

The top-level commands depend on the global state stored in GST module. If the global state is 0, GDP module gives commands of greeting. If the global state is 1, it generates commands to identify human identity. If the global state is 2, GDP module commands the system to generate responses.

The low-level commands vary based on the top-level commands. If the global state is 0 or 1, Scrip Bots module is assigned to generate response. If the global state is 2, Auto-biographical Memory module may be loaded, based on the state of human identification stored in GST module and both modules in Skills module may be commanded. Core Chat module may be asked to generate reponses contain memory words or not. Script Bots module may be required to generate reponses using retrieved facts or thoughts stored in GST module. Besides, Script Bots module is also able to suggest humans to change a story

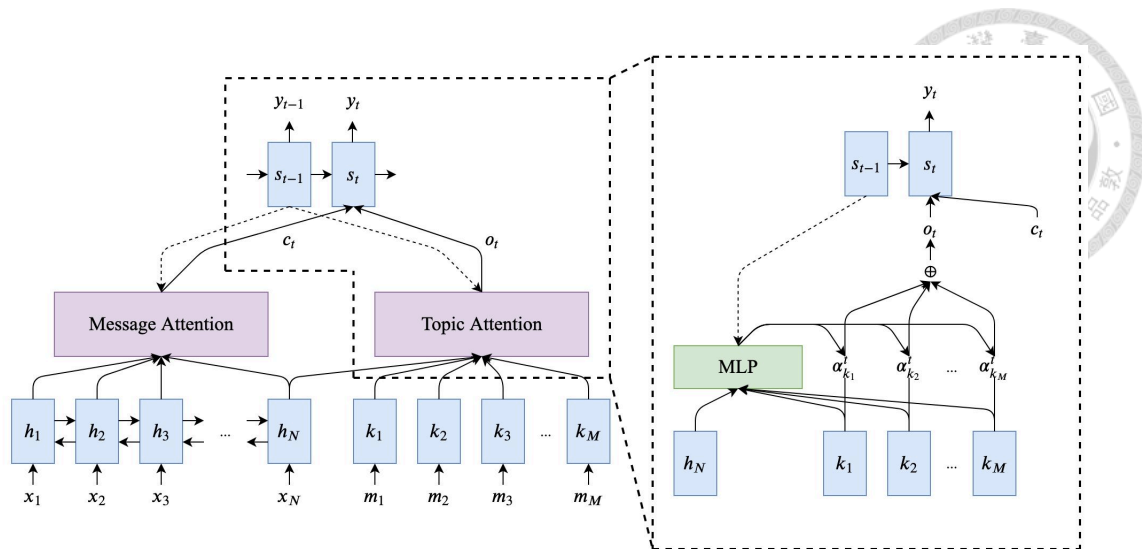


Figure 3.10: The architecture of Core Chat module (Adapted from [6]).

or a topic, if a sequence of negative sentiment is detected. At the end of the conversation, Script Bots module is responsible for saying ending words to humans.

3.5 Skills

Skills module takes the responsibility for generating responses according to human utterances. It can be divided into two submodules, namely Core Chat module and Script Bots module.

3.5.1 Core Chat

Core Chat takes most of the responsibility for generating robot's utterances with respect to human inputs. This module is designed to meet the requirement of generating both general responses as well as responses embedded with items retrieved from Memory KB. It's worth mentioning that if the identification of humans fails, then this module can only generate general responses.

In order to embed items from Memory KB into responses, the architecture of TA-

Seq2Seq model [6] is adopted, as shown in Figure 3.10. Given an input sequence $X = \{x_i\}_{i=1}^N$, a series of hidden vectors $\{h_i\}_{i=1}^N$ of them are generated through a bidirectional GRU (described in Section 2.5.2). At the same time, several memory words $M = \{m_j\}_{j=1}^M$ are transformed into hidden vectors $\{k_j\}_{j=1}^M$ through an embedding look-up table. At time step t , message attention, or attention mechanism, described in Section 2.6.1, generates a context vector c_t from $\{h_i\}_{i=1}^N$, which are the hidden vectors of the input sequence. A memory vector o_t is generated by memory attention, where each weight $\alpha_{k_j}^t$ for k_j is given as

$$\alpha_{k_j}^t = \frac{\exp(MLP(s_{t-1}, k_j, h_N))}{\sum_{j'=1}^M MLP(s_{t-1}, k_{j'}, h_N)} \quad (3.6)$$

where s_{t-1} is the previous hidden state in decoder and h_N is the last hidden state of the input sequence. Then s_t , the current hidden state in decoder is generated from c_t and o_t .

3.5.2 Script Bots

Script Bots module generates robot's responses through rule-based methods written in AIML (described in 2.2). It can be divided into three submodules, Memory Script Bot, Assistive Script Bot and Emotion Support Bot. These submodules generate robot's utterances obeying the commands from GDP module.

- **Memory Script Bot**

Memory Script Bot module is designed for generating robot's utterances embedded with items retrieved from Memory KB. The items belong to two categories, `fact` and `thought` from **Episodic Memory Layer**, since items from this two categories are hard to embed into sentences using ML-based dialogue modules. An example AIML code of

Memory Scriptbot is shown as follows.



```
<category>
  <pattern>*</pattern>
  <template>
    我記得你有說過<star/>
    <!--
    I remember that you have talked about <star/>. -->
  </template>
</category>
```

• *Assistive Script Bot*

Assistive Script Bot module is designed for three purposes: to greet, to identify users, and to say goodbye. For the greeting part, the module generates utterances actively to ask whether human is willing to chat with it, shown as follows. Then, if detected human intents are positive, responses like “終於有人陪我聊天了真開心” (Someone finally talk with me. So happy.) will be generated and the conversaiton will keep going, otherwise responses like “雖然有點難過再見” (Even though a little sad, bye.) will be generated and the conversation will stop. If human intents is ambiguous, the module generates responses like “不好意思你這是要和我聊天呢還是不要和我聊天呢” (Excuse me, would you like to chat with me or not?) to suggest human to make his/her intents clear.

```
<category>
  <pattern>start</pattern>
  <template>
    嗨 你好 我是胡椒 要和我聊聊天嗎
```

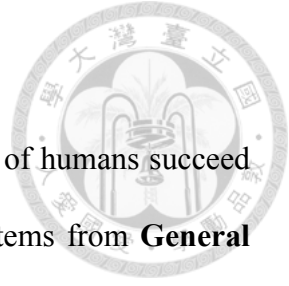


```
<!-- Hi. How are you? I'm Pepper.  
Would you like to chat with me? -->  
</template>  
</category>
```

For the part of identifying human identities, shown as follows, the module actively asks humans to tell their names. While human names have been told, the module confirms on human names. If human names are correct, then the process of identification is finished. For the case that humans refuse to tell their names, the module may generate responses like “好吧不勉強你想和我說什麼呢” (All right. No pressure. What do you want to talk to me?) and the state of identification on humans remains false. Moreover, if humans say something useless, in this stage, the module will actively say “好吧我放你你的名字了你想和我說什麼呢” (Well, I give up. What do you want to talk to me?) to skip the process of identifying humans and the state of human identification remains false.

```
<category>  
  <pattern>start</pattern>  
  <template>  
    我不確定之前是不是有和你聊過天 請告訴我你的名字  
    <!-- I'm not sure whether we have talked before.  
    Would you like to tell me your name please? -->  
  </template>  
</category>
```

The last task for Assitive Scrip Bot module is to say goodbye. As human intents to leave or to stop chatting is detected, the module generates “掰掰我會想你的下次聊” (See you. I will miss you. Chat with me next time.) to end conversations.



• *Emotion Support Bot*

Emotion Support Bot module works no matter the identification of humans succeed or fail. If the identification of humans succeed, the module picks items from **General Event Layer** in Memory KB which have large num_act to generate robot's utterances to suggest humans to change a topic or a story, shown as follows. The reason of choosing items with large num_act is because that the larger num_act an item has gained, the more familiar the item may be to humans.

```
<category>
  <pattern>*</pattern>
  <template>
    要不要換個話題 談談<star/> 如何
    <!-- Would you like to change a topic?
    What about <star/>? -->
  </template>
</category>
```

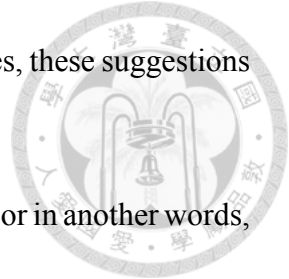
3.6 Natural Language Generation

NLG module is responsible for providing robot's responses to humans. In our design, NLG module provides humans with robot's responses in two ways. One is through texts and the other one is through graphs.

As for the part of texts, since multiple responses may be generated by Skills module, NLG module arranges the order of the generated responses. Responses generated by Core Chat module is always placed before responses generated by Script Bots module. If Script

Bots module is asked to generate suggestions to change topics or stories, these suggestions will be placed in at last.

Graphs also help the proposed system to perform robot's responses or in another words, robot's minds. Since not all the retrieved items from Memory KB will be embedded in generated sentences, the graphs full of nodes are used to illustrate them. One node is tagged with one retrieved item. Moreover, the probability of observing an retrieved item is indicated by the size of the node.





Chapter 4

Evaluation

In this chapter, evaluations of the proposed system is presented. Evaluations are divided into three parts, the evaluation of NLU module, the evaluation of Autobiographical Memory module, and the evaluation of Core Chat module.

4.1 Evaluation of Natural Language Understanding

The evaluation of NLU module is to show its ability to understand human utterances and extract essential memory information. Postings collected from social community are used as human utterances talking about their experiences in this experiment. After using NLU module to extract memory items from the collected postings, we ask humans to annotate these memory items. The annotation from humans are viewed as the ground truth for the evaluation of NLU module. The description of collected postings, experiment procedures, experimental results and discussions are shown in this section.

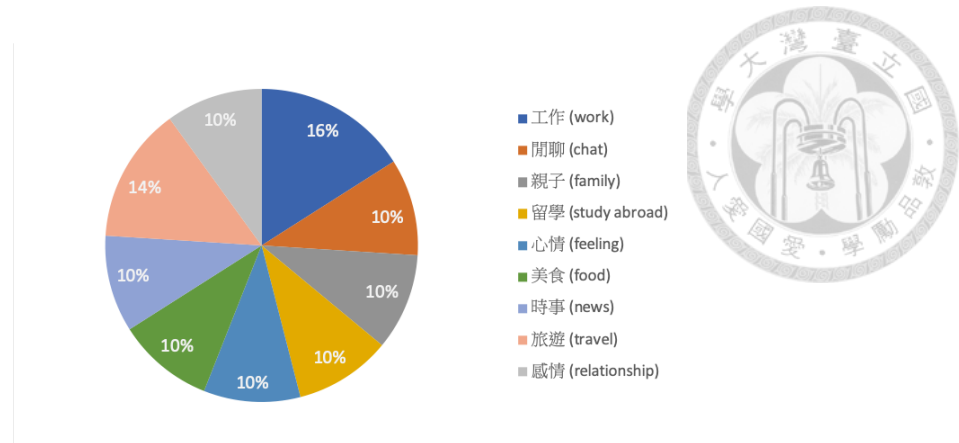


Figure 4.1: The distribution of collected postings over 11 categories.

4.1.1 Data Description

50 postings from Dcard¹ are collected, which is a social community that people can share their stories with each other. The postings spread over 11 categories, including “工作” (work), “閒聊” (chat), “親子” (family), “留學” (study abroad), “心情” (feeling), “美食” (food), “時事” (news), “旅遊” (travel) and “感情” (relationship), which follows the categories on Dcard. The distribution of these 50 postings are shown in Figure 4.1. The length of collected postings are in the range from 6 to 60 sentences.

4.1.2 Participants

We have invited 6 volunteers to participate in this experiment. There have been 4 males and 2 females, whose ages range from 23 to 26.

4.1.3 Procedure

After collecting postings from social community, NLU module is used to extract items which should be essential for the postings. The extracted items can be divided into five categories, including expression of human, location, activity & object, fact and time. Then

¹<https://www.dcard.tw/f>

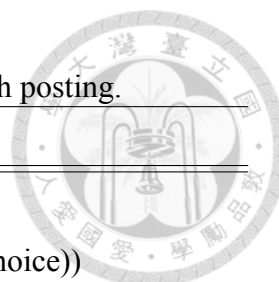


Table 4.1: Questions contained in the questionnaire for each posting.

Index	Question
1	此文章中出現過的主要人物有哪些?(可複選) (Which are the main characters in the posting? (Multiple choice))
2	此文章中出現過的主要地點有哪些?(可複選) (Which are the main locations in the posting? (Multiple choice))
3	此文章中出現過的主要事物(動名詞)有哪些?(可複選) (Which are the main activities or objects in the posting? (Multiple choice))
4	此文章中出現過的主要特別事物(被形容的動名詞)有那些?(可複選) (Which are the main special activities or objects (activities or objects with adjectives or adverbs) in the posting? (Multiple choice))
5	此文章中出現過的時間詞有哪些?(可複選) (Which are the main expressions of time in the posting? (Multiple choice))

each posting and its extracted items, along with designed questionnaire, are formed into a testing pair. Each questionnaire requires humans to read the posting first and then follow the designed questions to annotate memory items for the posting, based on their memory of the content.

For each posting, five questions are given to humans as instructions to help them do annotations. The questions are listed in Table 4.1. The extracted items are provided, as well as blankets for humans to add choices if necessary. An example of a testing pair is shown in Figure 4.2.

4.1.4 Results and Discussion

We use *Precision*, *Recall* and F_1 *Score* (containing both macro F_1 *Score* and micro F_1 *Score*) to analyze the performance of the proposed NLU module. The formula for



Posting

剛看完影片才發現
基隆可以體驗到鐵道自行車
從八斗子車站可慢慢騎到深奧車站
沿途還可以吹海風看風景
車子是可愛的海豚造型
中途還有星海隧道
感覺這會很適合小情侶
我女友看完影片就開始吵著要去了

Instruction

本實驗為Event Extraction的檢測
共十篇文章
會花一點時間
每一篇文章都會搭配相應的問題
請大家仔細閱讀完文字後記住文章的內容
再前往下一頁進行作答
作答期間請勿回到上一頁對照文章

Questionnaire

1. 此文章中出現過的主要人物有那些？(可複選)
星海 女友 其他：_____
2. 此文章中出現過的主要地點有那些？(可複選)
八斗子 其他：_____
3. 此文章中出現過的主要事物(動詞/名詞)有那些？(可複選)
看完影片 體驗到隧道 看影片
到鐵道自行車 騎到深 到深奧
體驗到 其他：_____
4. 此文章中出現過的主要特別事物(被形容的動詞/名詞)有那些？(可複選)
剛看 可愛的海豚 才發現 慢慢騎
很適合 小情侶 其他：_____
5. 此文章中出現過的時間詞有那些？(可複選)
無 其他：_____

Figure 4.2: An example of the testing pair for the evaluation of NLU module.

Table 4.2: The *Precision*, *Recall* and F_1 *Score* of NLU module of different categories.

Category	human	location	activity & object	fact	time
<i>Precision</i>	0.833	0.667	0.587	0.525	0.972
<i>Recall</i>	0.843	0.884	0.981	0.978	0.814
F_1 <i>Score</i>	0.838	0.760	0.734	0.683	0.886

Precision, *Recall* and F_1 *Score* is

$$\begin{aligned}
 Precision &= \frac{TP}{TP + FP} \\
 Recall &= \frac{TP}{TP + FN} \\
 F_1 \text{ score} &= \frac{Precision \cdot Recall}{Precision + Recall}
 \end{aligned}
 \tag{4.1}$$

where TP denotes true positive, FP denotes false positive and FN denotes false negative.

In this experiment, items chosen by both humans and NLU modules are viewed as true positive. Items chosen only chosen by NLU modules are viewed as false positive, and items added by humans are viewed as false negative. The results are shown in Table 4.2 and Table 4.3, where one shows the results of each category and the other one shows the results of all postings.



Table 4.3: The *Precision*, *Recall* and F_1 *Score* of NLU module of all postings.

Category	marco	micro
<i>Precision</i>	0.717	0.612
<i>Recall</i>	0.900	0.935
F_1 <i>Score</i>	0.790	0.740

In Table 4.2, it's obvious that NLU module gains higher *Precision* on the categories of human, location and time. One reason is that expressions of categories of human, location and time are not that various as expressions of categories of activity & object and fact. Therefore, when extracting items for categories of human, location and time, it's much more possible that the extracted items are correct. Another reason is that expressions of categories of human, location and time are much more specific in a story, in contrast, expressions of categories of activity & object and fact are not that distinct, which increases the probability of extracting something trivial or nonsense.

NLU module seems to gain pretty high *Recall* in all categories. It's worth noticing that items that can be memorized by humans are limited, so that it's possible that humans are not able to recall all essential items from postings, which limits the number of human-added items when doing annotation.

In Table 4.3, it can be observed that there's not much difference between marco and micro *Precision*, *Recall* and F_1 *Score*, where micro scores are more suitable for the case when postings are not equally distributed in chosen categories. Therefore, we can claim that the proposed NLU module is suitable for extracting items from postings from various categories.

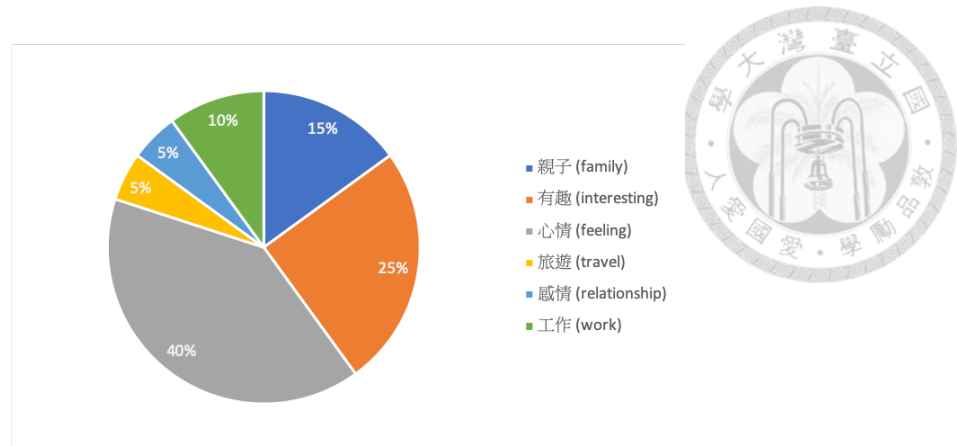


Figure 4.3: The distribution of collected postings over 6 categories.

4.2 Evaluation of Autobiographical Memory

The evaluation of Autobiographical Memory module is to show the performance of proposed Memory KB as well as the ability of Memory DBMS to retrieve relevant memory items from Memory KB. Postings collected from social community are used as human memories and are stored into Memory KB. Summaries for postings are prepared and are used to retrieve relevant memory items from Memory KB. The description of collected postings, summaries of the postings, experiment procedures, experimental results and discussions are shown in this section.

4.2.1 Data Description

20 postings from Dcard are collected. The postings spread over 6 categories, including “親子” (family), “有趣” (interesting), “心情” (feeling), “旅遊” (travel), “感情” (relationship) and “工作” (work), which follows the categories on Dcard. The distribution of these 20 postings are shown in Figure 4.3. The length of collected postings are in the range from 7 to 29 sentences.

In order to retrieve items of collected postings from Memory KB, each posting is prepared with a summary written by humans. An example of the posting and its summary

Posting

早上騎車開開信心的出門
結果撞到一台急左轉的娃娃車
在等警察來的時候我發現一家早餐店
突然覺得蛋餅好香好想吃
警察先生對我測完酒測後
我就說我先去吃早餐喔等交通隊來我
再過來
警察先生的表情彷彿在告訴我
你都出車禍了你還會想吃早餐
不過真的幸好小朋友們都沒受傷
還很開心的在跟我打招呼
我也很開心的對他們打招呼
喂出車禍好像不該這麼開心欸

Written Summary

有一天早上我開開心心騎機車出門
結果和一台娃娃車發生車禍
在等警察來的時候發現一間感覺很好吃的早餐店
然後警察來了之後我想說先去買早餐再回來應該可以吧
警察一副不可思議的表情
而且我還和娃娃車上的小朋友們開心的打招呼
出車禍這樣對嗎



Figure 4.4: An example of the posting and its summary.

is shown in Figure 4.4.

4.2.2 Procedure

After collecting postings from social community, NLU modules is firstly used to extract memory items for the postings. Then the experiment are conducted in two ways.

In the first experiment, we aims to test the performance of Autobiographical Memory module under low memory density. In this experiment, two posting and a summary of one of the posting form into a testing pair. Both the postings are stored into Memory KB, and the summary is used to retrieve relevant memory items from Memory KB.

In the second experiment, we aims to test the performance of Autobiographical Memory module under high memory density. In this experiment, all postings are stored into Memory KB. Each of the summary is used to retrieve relevant memory items from Memory KB.

In order to find a suitable *Threshold* (mentioned in Algorithm 1) for the procedure of retrieval, three values of *Threshold* are tested in both experiments.



Table 4.4: Results of Autobiographical Memory module under low memory density.

<i>Threshold</i>	1	3	5
<i>Precision</i>	0.761	0.95	1.0
<i>Recall</i>	0.811	0.804	0.283
<i>F₁ Score (marco)</i>	0.785	0.871	0.411

4.2.3 Results and Discussion

We use *Precision*, *Recall* and *F₁ Score* (marco *F₁ Score*) to analyze the performance of the proposed Autobiographical Memory module. The formula for *Precision*, *Recall* and *F₁ Score* is shown in Equation 4.1. The results of the first experiment are shown in Table 4.4, which shows the performance of Autobiographical Memory module under low memory density.

In Table 4.4, it can be observed that the lower the *Threshold* is, the higher the *Recall* will be, however, with lower *Precision*. The reason is that if the *Threshold* is set to be low, it's more likely for Autobiographical Memory module to consider the written summary being relevant to both postings. For example, if an item from the summary occurs in both postings, under the situation that the *Threshold* is set to be 1, then items from both postings will be retrieved, even though the summary belongs to only one of the postings. In contrast, the higher the *Threshold* is, the higher the *Precision* will be, but with lower *Recall*. It's also intuitive that high *Threshold* means the requirement of high repetition of items from the summary and the relevant posting. Due to the requirement of high repetition, the *Precision* gets higher, but the *Recall* becomes lower.

The results of the second experiment are shown in Table 4.5, which shows the performance of Autobiographical Memory module under high memory density.

In Table 4.5, it's obvious that the performance of Autobiographical Memory module



Table 4.5: Results of Autobiographical Memory module under high memory density.

<i>Threshold</i>	1	3	5
<i>Precision</i>	0.025	0.468	0.356
<i>Recall</i>	0.045	0.472	0.288
<i>F₁ Score (marco)</i>	0.032	0.470	0.319

under high memory density are not as well as under low memory density. Several reasons may lead to this situation. One main reason is that if the density of memory stored in Memory KB increases, items from similar postings will be retrieved. Therefore, it results in the decrease of *Precision* and *Recall*. Another reason is that with the increase of memory density makes it more possible for Memory DBMS to view similar postings as the same posting, and it's even worse if postings don't obtain distinct items. From one aspect, it can be claimed that the performance of Autobiographical Memory module under high memory density get worse, but from another aspect, it increases the probability for Autobiographical Memory module to retrieve similar or relevant memories from Memory KB to help human-robot conversations.

From both experiments, *Threshold* of 3 seems to be much more suitable than other two values. Therefore, *Threshold* of 3 is adopted in our system. Moreover, we conduct another experiment to show the situation of *Recall* for *Threshold* of 3 in detail. The experiment result is shown in Figure 4.5. The darker the block is, the higher the *Recall* is. The best case is that the diagonal are dark and others are light. Even though in most of the case dark blocks appear on the diagonal, in some cases, green blocks appear in places other than the diagonal, which reduces *Recall* of *Threshold* of 3.

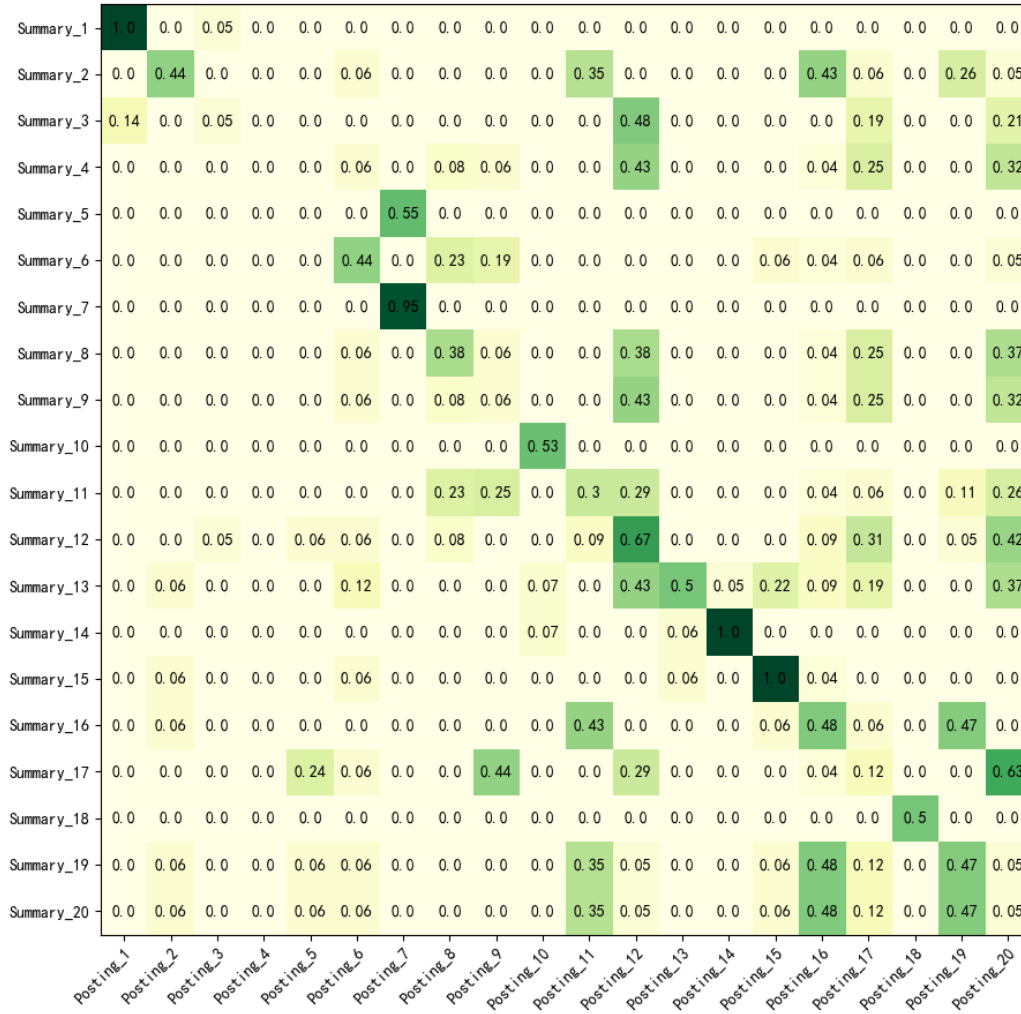
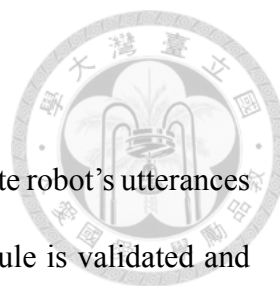


Figure 4.5: The experiment result of *Recall* for *Threshold* of 3.



4.3 Evaluation of Core Chat

The evaluation of Core Chat module is to show its ability to generate robot's utterances embedded with memory items. The performance of Core Chat module is validated and is compared with a Seq2Seq with Attention model. Besides, human evaluations are also conducted. The description of data used for training, validation and testing, parameters of model, experimental results and discussions are shown in this section.

4.3.1 Data Description

We use Douban Conversation Corpus [30] to train, valid and test Core Chat module. Douban Conversation Corpus is a multi-turn conversation dataset collected for retrieval-based response generation, which contains 1 million training data, 50 thousand validation data and 50 thousand testing data. After removing fake data, only half of the data is left. Then multi-turn contexts are separated into single turn pairs. Among all single turn pairs, pairs that contain sentences which are longer than 25 words are removed. Besides, English characters and numerical numbers are removed as well. After that, we reorganize the data and obtain 1,440,136 training pairs, 75,677 validating pairs and 1,893 testing pairs. A threshold is set to 30 to build the vocabulary for both input and output sentences, and 25,979 words are left in the vocabulary. The input memory words are extracted from both queries and responses through our proposed NLU module. The reason of that is to let our model learn to generate sentences using given words but to not too rely on given words. The given words belong to categories of human, location, activity & object, and fact. Some example of training, validation and testing data are shown in Table 4.6.



Table 4.6: Examples of data used to train, valid and test Core Chat module.

Input Utterance:	不能用銀聯是要用臉刷卡麼
Input Memory Items:	單銀聯買不起刷能卡應該
Ground Truth:	不能用銀聯的地方應該買得起單 <eos>

Input Utterance:	跟瓶女在一起需要學會欲擒故縱
Input Memory Items:	畫跳瓶女一起女學會好想圈需要
Ground Truth:	只要瓶女想要的她一定會畫好圈讓你跳的 <eos>

Input Utterance:	挖牆腳不成
Input Memory Items:	不成喜歡挖送牆角基友
Ground Truth:	你喜歡就送你吧反正我基友一大堆 <eos>

4.3.2 Parameters

In Core Chat module, the maximum number of input memory items is set to 10. The embedding dimension is set to 256, the encoder dimension and the decoder dimension are set to 512, and both attention dimensions are set to 128. The model is trained using Adam optimizer [46] with learning rate equal to 0.0001 on NVIDIA GeForce 1080 GPU. The batch size is 128. We implement the model with an open source deep learning tool TensorFlow².

4.3.3 Results and Discussion

Since most of the evaluation metrics are not able to represent human judgements on responses generated by a dialogue system [47], in this experiment, the evaluation of Core Chat module mainly relies metrics defined for human evaluations. Firstly, to prove that our model has learned from the training data, the training loss and validation loss is shown in Figure 4.6 and Figure 4.7. It can be observed that the model has converged.

In Table 4.7, we compute the perplexity for validation and testing data. The perplexity

²<https://www.tensorflow.org/>

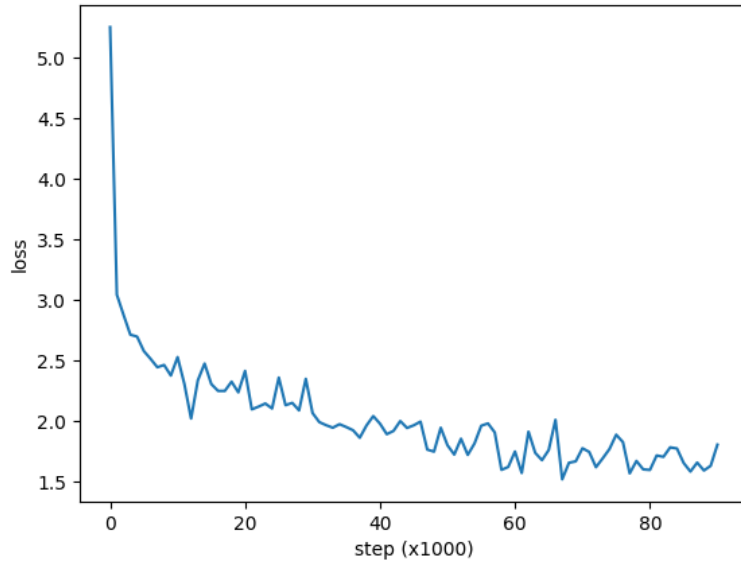


Figure 4.6: The training loss of Core Chat module.

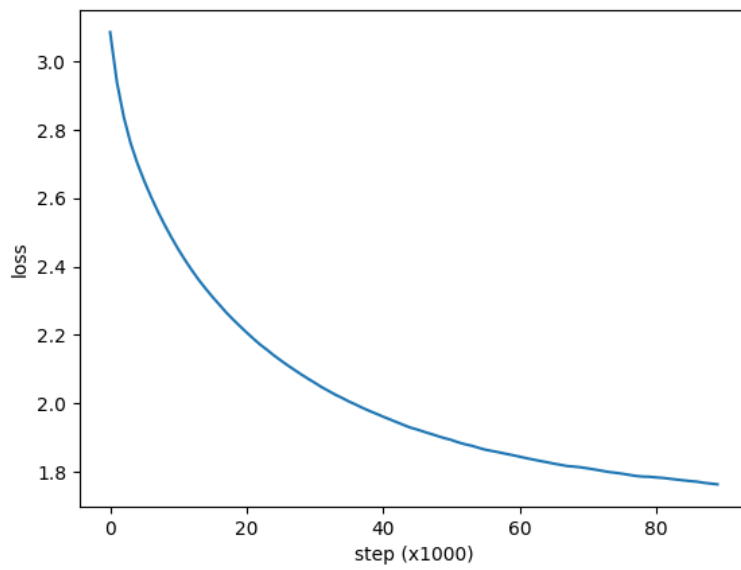


Figure 4.7: The validation loss of Core Chat module.

Table 4.7: The perplexity of Core Chat module compared with Seq2SeqAtt.

	PPL_D	PPL_T
Seq2SeqAtt	27.132	26.851
Core Chat (ours)	5.82	5.47



of validation and testing data are denoted as PPL_D and PPL_T respectively, and they are compared with those of a Seq2Seq with Attention (Seq2SeqAtt) model. The results show that our model learned better than the baseline model.

Since perplexity are not fully correlated to human judgements, we designed three metrics for humans to judge the performance of Core Chat module. The first metric is to evaluate the correctness of generated responses on grammar, and is defined as

$$G(S) = \frac{1}{N} \sum_{i=1}^N g(s_i), s_i \in S$$

$$g(s_i) = \begin{cases} 1, & \text{if grammar is correct} \\ 0, & \text{if grammar is wrong} \end{cases} \quad (4.2)$$

where S denotes a set of testing sentences.

The second metric is to evaluate the ability of embedded memory items into generated responses, and is defined as

$$M(S) = \frac{1}{N} \sum_{i=1}^N m(s_i), s_i \in S$$

$$m(s_i) = \begin{cases} 1, & \text{if contains memory items} \\ 0, & \text{if not contains memory items} \end{cases} \quad (4.3)$$

The third metric is to evaluate the relatedness of generated sentences with input sen-

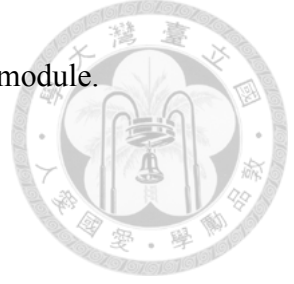


Table 4.8: The result of human judgement for Core Chat module.

$G(S)$	$M(S)$	$R(S)$
0.70	0.98	0.84

tences, where it is defined as

$$R(S) = \frac{1}{N} \sum_{i=1}^N r(s_i), s_i \in S$$
$$r(s_i) = \begin{cases} 1, & \text{if related to inputs} \\ 0, & \text{if not related to inputs} \end{cases} \quad (4.4)$$

50 sentences are sampled from testing data to form the evaluation set S . The results of human judgement is shown in Table 4.8. It's worth noticing that beam search is used during the generation of sentences and the beam size is set to 5. From the results, we can conclude that generated sentences have weak grammar which are not pleasant enough to chat with humans, but they do embed input memory items into sentences and they are somehow related to input sentences. One of the reason why the correctness of grammar is low is because of the dataset. Since the dataset comes from the Internet, it contains expressions not suitable to use in oral conversations. Moreover, utterances collected from the Internet contain various abbreviations and combinations of multiple sentences, which increase the difficulty for Core Chat module to learn to generate robot's responses. Some of the generated responses are shown in Table 4.9



Table 4.9: Examples of responses generated by Core Chat module.

Input Utterance:	還有到了市鎮給小孫子買巧克力那一段
Input Memory Items:	到買名字小劇忘記總是巧克力孫子
Ground Truth Responses:	但是我總是忘記這劇的名字
Generated Responses:	我也忘記了這劇的名字總是忘記了

Input Utterance:	啊啊啊每天便便順暢可是還是得了痔瘡啊
Input Memory Items:	淚得痔瘡說
Ground Truth Response:	同說多了都是淚啊
Generated Response:	我說的是淚啊

Input Utterance:	看照片他就染了的頭髮捏挑染省點吧
Input Memory Items:	認識不捏媽辦看染完萬一照片
Ground Truth Response:	萬一染完我媽都不認識我了咋辦啊
Generated Response:	萬一我媽不認識我我就不辦了

Input Utterance:	我多想找到你然後一起去旅行
Input Memory Items:	找到先嗎旅行想多
Ground Truth Response:	可以只是旅行嗎先
Generated Response:	先去旅行吧

Input Utterance:	從今天開始抽個小時學習英語口語
Input Memory Items:	抽學習練英語口語今天人
Ground Truth Response:	一個人怎麼練呢
Generated Response:	我也是練口語的人



Chapter 5

Conclusion

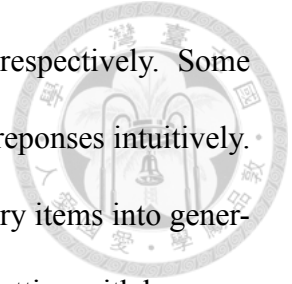
In this thesis, a dialogue system integrated with a computational memory model is proposed. The main purpose is to provide robots with the ability to memorize while communicating with humans. Through the assistance offered by the computational memory model, robots are able to build better relationship with humans, which may help in the field of elderly care.

Three objectives are fulfilled in this work. Firstly, a computational memory model is built as well as its DBMS. The computational memory model, Autobiographical Memory module, is built upon a conceptual autobiographical memory model. Memory KB in Autobiographical Memory module uses a four-layer structure to accomplish the storage of both episodic knowledge and semantic knowledge, which can be viewed as the identity of humans. Memory DBMS helps to retrieve memory items from Memory KB which are relevant to human utterances, and also helps to update new memory items into Memory KB. Secondly, a NLU module is built, which tries to understand human utterances and extracts memory items. Thirdly, Skills module is proposed to embed memory items into robot's utterances through both rule-based methods and generation-based methods. Moreover, a

DM module, including GST and GDP, is proposed to help to preserve information needed in a dialogue session and control the workflow of overall dialogue system. Besides, a NLG module helps to present robot's utterances and robot's mind to humans.

The evaluation of the proposed system contain four parts. Firstly, NLU module is evaluated using postings collected from social community. The extracted items are compared with human annotations and metrics of precision, recall and F1 Score are used to analyze. The experiment results show that the proposed NLU module achieves precision of 71.7%, recall of 90.0%, and F1 Score of 79.0%, which indicates the ability of NLU module to extract memory items from human utterances. Secondly, Autobiographical Memory module is evaluated using postings collected from social community as well. Each collected posting is provided with a human-written summary, which is used to retrieve relevant posting items from Memory KB. Metrics of precision, recall and F1 Score are used in this experiment as well. The experiment can be divided into two parts, low memory density Memory KB and high memory density Memory KB. Under low memory density, Autobiographical Memory module performs well, achieving precision of 95.0%, recall of 80.4% and F1 Score of 87.1% when *Threshold* equals to 3. However, under high memory density, the performance of Autobiographical Memory module is not pleasant enough. It achieves precision of 46.8%, recall of 47.2% and F1 Score of 46.0%, which indicates an improvement on Autobiographical Memory module is necessary. Thirdly, Core Chat module is evaluated using Douban Conversation Corpus. The plots of losses show the convergence of Core Chat module and the perplexity of validation data and testing data is computed. It achieves perplexity of 11.341 and 11.217 on the validation data and testing data respectively. Human judgement are also conducted to examine the correctness of grammar, the ability to embed memory items into sentences and the relatedness to input sentences.

After human judging, Core Chat module achieves 0.34, 0.86, 0.98, respectively. Some generated examples are listed to show the performance of generated responses intuitively. Even though generated responses have learned to embed input memory items into generated sentences, the generated sentences are not satisfied enough for chatting with humans, that is, a better model may be needed. Finally, the overall system is evaluated through human-robot interaction experiment.





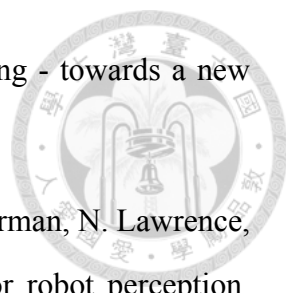
References


- [1] M. A. Conway, “Memory and the self,” *Journal of Memory and Language*, vol. 53, no. 4, pp. 594 – 628, 2005.
- [2] J. Weizenbaum, “Eliza - a computer program for the study of natural language communication between man and machine,” *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.
- [3] O. Vinyals and Q. V. Le, “A neural conversational model,” *CoRR*, vol. abs/1506.05869, 2015.
- [4] J. Williams, A. Raux, and M. Henderson, “The Dialog State Tracking Challenge Series: A Review,” *Dialogue and Discourse*, April 2016.
- [5] H. Fang, H. Cheng, M. Sap, E. Clark, A. Holtzman, Y. Choi, N. A. Smith, and M. Ostendorf, “Sounding Board: A User-Centric and Content-Driven Social Chatbot,” *arXiv e-prints*, p. arXiv:1804.10202, Apr 2018.
- [6] C. Xing, W. Wu, Y. Wu, J. Liu, Y. Huang, M. Zhou, and W.-Y. Ma, “Topic aware neural response generation,” 2017.
- [7] S. Yusif, J. Soar, and A. Hafeez-Baig, “Older people, assistive technologies, and the barriers to adoption: A systematic review,” *International Journal of Medical Informatics*, vol. 94, pp. 112 – 116, 2016.
- [8] United Nations, 2015.
- [9] S. Vichitvanichphonng, A. Talaei-Khoei, D. Kerr, and A. H. Ghapanchi, “Assistive technologies for aged care: Comparative literature survey on the effectiveness of


theories for supportive and empowering technologies,” *Information Technology & People*, vol. 31, no. 2, pp. 405–427, 2018.



- [10] R. Wilkinson and M. Marmot. World Health Organization, 2003.
- [11] S. Westwood and M. Daly, “Social Care and Older People in Home and Community Contexts: A Review of Existing Research and Evidence,” 2016.
- [12] M. J. Broekens and H. Rosendal, “Assistive social robots in elderly care: a review,” *Gerontechnology*, pp. 94–103, 2009.
- [13] M. Fujita, “AIBO: Toward the Era of Digital Creatures,” *The International Journal of Robotics Research*, vol. 20, no. 10, pp. 781–794, 2001.
- [14] A. M. TURING, “I.—COMPUTING MACHINERY AND INTELLIGENCE,” *Mind*, vol. LIX, pp. 433–460, 10 1950.
- [15] R. C. Atkinson and R. M. Shiffrin, “Human memory: A proposed system and its control processes,” *Psychology of Learning and Motivation*, vol. 2, pp. 89–195, 1968.
- [16] A. Baddeley, vol. 45 of *Oxford psychology series*. New York, NY, US: Oxford University Press, 2007.
- [17] M. A. C. Helen L. Williams and G. Cohen. 2008.
- [18] E. Tulving, “Episodic and semantic memory,” 1972.
- [19] M. A. Conway and C. W. Pleydell-pearce, “The construction of autobiographical memories in the self-memory system,” *Psychological Review*, pp. 261–288, 2000.
- [20] W. H. B. A. D. Conway, M. A., “Boundaries in episodic memories: some initial findings,” 2005.
- [21] G. Pointeau, M. Petit, and P. F. Dominey, “Successive developmental levels of autobiographical memory for learning through social interaction,” *IEEE Transactions on Autonomous Mental Development*, vol. 6, pp. 200–212, Sep. 2014.

- 
- [22] M. H. Evans, C. W. Fox, and T. J. Prescott, “Machines learning - towards a new synthetic autobiographical memory,” pp. 84–96, 2014.
- [23] U. Martinez-Hernandez, A. Damianou, D. Camilleri, L. W. Boorman, N. Lawrence, and T. J. Prescott, “An integrated probabilistic framework for robot perception, learning and memory,” pp. 1796–1801, Dec 2016.
- [24] W. C. Ho and K. Dautenhahn, “Towards a narrative mind: The creation of coherent life stories for believable virtual agents,” pp. 59–72, 2008.
- [25] W. C. Ho, K. Dautenhahn, M. Y. Lim, P. A. Vargas, R. Aylett, and S. Enz, “An initial memory model for virtual and robot companions supporting migration and long-term interaction,” pp. 277–284, Sep. 2009.
- [26] J. Campos and A. Paiva, “May: My memories are yours,” pp. 406–412, 2010.
- [27] L. Zhou, J. Gao, D. Li, and H.-Y. Shum, “The Design and Implementation of Xiaolce, an Empathetic Social Chatbot,” *arXiv e-prints*, p. arXiv:1812.08989, Dec 2018.
- [28] A. Ritter, C. Cherry, and W. B. Dolan, “Data-driven response generation in social media,” pp. 583–593, 2011.
- [29] Z. Yan, N. Duan, J. Bao, P. Chen, M. Zhou, Z. Li, and J. Zhou, “Docchat: An information retrieval approach for chatbot engines using unstructured documents,” pp. 516–525, 01 2016.
- [30] Y. Wu, W. Wu, C. Xing, M. Zhou, and Z. Li, “Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots,” pp. 496–505, july 2017.
- [31] L. Shang, Z. Lu, and H. Li, “Neural responding machine for short-text conversation,” pp. 1577–1586, july 2015.
- [32] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J. Nie, J. Gao, and B. Dolan, “A neural network approach to context-sensitive generation of conversational responses,” *CoRR*, vol. abs/1506.06714, 2015.

- 
- [33] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, “A diversity-promoting objective function for neural conversation models,” pp. 110–119, june 2016.
- [34] J. Gu, Z. Lu, H. Li, and V. O. K. Li, “Incorporating copying mechanism in sequence-to-sequence learning,” *CoRR*, vol. abs/1603.06393, 2016.
- [35] E. F. Codd, “A relational model of data for large shared data banks,” *Commun. ACM*, vol. 13, pp. 377–387, June 1970.
- [36] M. Schuster, K. K. Paliwal, and A. General, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, 1997.
- [37] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [38] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” pp. 1724–1734, October 2014.
- [39] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv e-prints*, p. arXiv:1409.0473, Sep 2014.
- [40] M.-T. Luong, H. Pham, and C. D. Manning, “Effective Approaches to Attention-based Neural Machine Translation,” *arXiv e-prints*, p. arXiv:1508.04025, Aug 2015.
- [41] X. Li, Y.-N. Chen, L. Li, J. Gao, and A. Celikyilmaz, “Investigation of Language Understanding Impact for Reinforcement Learning Based Dialogue Systems,” *arXiv e-prints*, p. arXiv:1703.07055, Mar 2017.
- [42] J. Piskorski and R. Yangarber, “Information extraction: Past, present and future,” pp. 23–49, 2013.
- [43] H. Xu and C.-R. Huang, “A rule system for Chinese time entity recognition by comprehensive linguistic study,” pp. 795–801, oct 2013.

- 
- [44] J. Xia, F. Xie, M. Zhang, Y. Su, and H. Luan, “Cnme: A system for chinese news meta-data extraction,” pp. 91–107, 03 2016.
- [45] L.-W. Ku, H.-W. Ho, and H.-H. Chen, “Opinion mining and relationship discovery using copeopi opinion analysis system,” *JASIST*, vol. 60, pp. 1486–1503, 07 2009.
- [46] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv e-prints*, p. arXiv:1412.6980, Dec 2014.
- [47] C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, “How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation,” *arXiv e-prints*, p. arXiv:1603.08023, Mar 2016.