A REINFORCEMENT LEARNING MULTI-AGENT SYSTEMS ARCHITECTURE FOR GUARANTEEING ADVANCING CONVERSATIONS IN TASK-ORIENTED DIALOG SYSTEMS

BY

KEVIN MUGOYE SINDU

A THESIS SUBMITTED IN FULFILLMENT OF REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

SCHOOL OF COMPUTING AND INFORMATICS

MASENO UNIVERSITY

DECLARATION

I, Kevin Mugoye Sindu, do hereby declare that this PhD research is entirely my own work and where there's work or contributions of other individuals, it has been duly acknowledged. To the best of my knowledge, this dissertation has not been previously presented to any other education institution or for any other academic award.

Signature ___________________________ Date ____________________

Kevin Mugoye Sindu
Maseno University

I, **Dr. Henry Okora Okoyo**, do hereby certify that this PhD research has been presented for the award of Doctor of Philosophy with my approval as the Maseno University Supervisor.

Signature ___________________________ Date ____________________

Dr. Henry Okora Okoyo
Department of Computer Science
Maseno University.

I, **Dr. Sylvester O McOyowo**, do hereby certify that this PhD research has been presented for the award of Doctor of Philosophy with my approval as the Maseno University Supervisor.

Signature ___________________________ Date ____________________

Dr. Sylvester O McOyowo
Department of Computer Science
Maseno University.

ACKNOWLEDGEMENT

Firstly, I would like to acknowledge God Almighty, for providing me all that I required to pursue this PhD program.

I would like to express my sincere gratitude to my Supervisors Dr. Henry Okoyo and Dr. Sylvester McOyowo for the continuous support of my PhD study and related research, for their patience, motivation, and immense knowledge. Our meetings and seminars made this thesis possible. I would like to thank all who participated in evaluating the prototype and filling out the survey. Besides, I am grateful to the Maseno SCI fraternity and my fellow doctoral students for their patience, encouragement, insightful comments, cooperation and of course friendship. Finally, thanks to my family for supporting me spiritually throughout writing this thesis and my life in general.

DEDICATION

To my lovely wife, Edna; You have endured the long nights and absences when research severally took me away from bed and home. Thank you for the love and support throughout this journey

To my son, Jabali; Thanks for standing with me all the time despite missed days and plays. You're my treasure beyond compare.

ABSTRACT

The conversational capabilities of a dialog system have a direct impact on the tasks it can accomplish. Solving all conversational issues in dialog systems have the potential to make them serve in complex domains. While this is not achievable, addressing fundamental aspects in conversation is desired to make a task-oriented dialog system (TODS) serve in new domains where they are needed, besides increasing their usefulness. One such aspect is the ability to advance a conversation logically. The primary aim of this study was to develop a novel architecture that will guarantee advancing conversations in TODS. To realize this aim, theories and literature were interrogated that informed the formulation of an agent-based architecture for dialog management. Then implementation of the architecture previously realized in a dialog system prototype. Followed by training the dialog system on initial domain-specific data. And evaluating its performances in a specific domain. The study used exploratory methodology to provide the theories that justified the construction of the multi-agent system (MAS_DM) architecture, while the experimental design was explored to synthesize and train the prototype. The design involved the fusion of agent-based architecture with reinforcement learning technique to enable tracking of context, structure and policy without depending on handcrafted rules. MAS_DM architecture explores learning agents in an unknown environment, where each agent is endowed with the ability to learn and select a policy. Learning and policy selection is sustained through reinforcement learning, eliminating the need for handcrafted rules. The architectural model was evaluated and validated in a prototype Chabot system. The Chatbot system was trained and tested in the maternal healthcare domain and was evaluated by human users. In this context, each user filled out an online questionnaire after successful interaction with the Chatbot. The evaluation parameters were coherence, task success, general performance, user satisfaction and goal achievement. This evaluation adheres to the specifications of Goal Question Metrics and PARAdigm for DIalog System Evaluation frameworks. The key findings were that Chabot's ability to advance the conversation scored 0.8903, and achieved an overall performance score of 0.553. It achieved a task success rate of 0.936. with a user satisfaction score of 0.775. Based on global acceptable measures, interpreted this task success as substantial, coherence score as substantial, user satisfaction as excellent and the overall performance as good. Where machine learning is involved kappa statistic values above 0.40 are considered exceptional. The results suggest that it is reasonable to conclude that the MAS_DM architecture can be trusted to guarantee conversations that advance logically. The study contributes to the body of knowledge of conversational artificial intelligence by; - developing a novel agent-based architectural model for TODS, demonstrating the practicability of combining multi-agent systems and machine learning toward solving conversational issues and enhancing the capability of TODS.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

AI …………..….	Artificial Intelligence
ANOVA ……….	Analysis of Variance
API ………….....	Application Programming Interface
AVM ……..…..	Attribute Value Matrix
BDI ………….....	Belief-Desire-Intention
CODS …………	Chat Oriented Dialog System
CS ……….……..	Conversational software
CUI …………..….	Conversational User Interface
DA……………..….	Dialog Agent
DAI …………..….	Distributed Artificial Intelligence
DM ………….….	Dialog Manager
DMA …………...	Dialog Management Architecture
DRL ………..….	Deep reinforcement learning
DS ……………..….	Dialog System
GQM ………….	Goal Question Metric
GUI …………..….	Graphical User Interface
ISO …………..….	International Organization for Standardization
KB …………….	Knowledge Base
KB …………..….	Knowledge Base
MA …………….	Master agent
MAS ………..….	Multi-agent System
MAS_DM …..….	Multi-agent Systems based dialog management
MDP ………..….	Markov Decision Processes
NLU ………..….	Natural Language Understanding
PARADISE …....	PARAdigm for DIalog System Evaluation
RA ……….…….	RL Agent
RL ……..……….	Reinforcement Learning
SA ……..……….	State Agent
SAS ……...…….	Single-Agent System
TODS ……….....	Task-Oriented Dialog System
WM ……………	Working Memory

List of Tables

LIST OF FIGURES

LIST OF APPENDICES

GLOSSARY

Adaptive:	The term "adaptation" in our context refers to a process, in which an interactive system (**adaptive** system) adapts its behaviour to individual users based on information acquired about its user(s) and its environment.
Advancing conversation:	A progression of exchanges which make reference to previous statement in the current situation – that allows grows within a context. Normally intended to achieve a goal.
Coherence:	Indicates the relevance and how comprehensible a response is to a user's request. A metric for measuring conversational depth.
Dialog system:	A conversational software with which a user can converse in a natural language.
Generality:	A measure of how "human-friendly" and easy to learn, use and memorize a language is.
Goal oriented action:	In our context refers system that allows for agents to execute actions to satisfy a particular goal. The particular action(s) depends not only on the goal but also on the current state of the world and the agent.
Handcrafts:	handcrafts or handcrafted rules, refer to additional algorithms, auxiliary statistical rules, auxiliary data structures added or appended to the architecture to improve the performance or add a missing functionality in the dialog system. The rules are separately appended to the system
Intelligence:	Indicates that agents pursue their goals and execute their tasks such that they optimize some given performance measures.
Open system:	The term "open" means that the model takes also care of the environment, which can influence the behaviour of the overall system directly.
Robustness:	The ability of a computer system to cope with errors during execution and cope with erroneous input.
Structure:	Describes the relation between concepts and exchanges.
Text in-text out TODSs:	Describe interfaces whose interaction is supported via text.
TODS:	Task oriented dialog system.
CODS:	Chat oriented dialog system.

CHAPTER ONE

INTRODUCTION

This chapter defines and classifies dialog systems, then presents possible scenarios where the capability of dialog systems seems to be challenged. In the subsequent sections it suggests a novel approach to improve the effectiveness of dialog system followed by the attempt to justify why dialog systems may not be of service to some new domains. It then presents the objective of the study, the research contribution, scope and significance, and concludes the chapter with the assumption and limitations of the study.

1.1 Background of the study

A dialog system (DS) is conversational software with which a user can converse in a natural language such as English or Kiswahili. It provides a very good tool for interaction between any application and user. This thesis divided dialog systems into two broad categories, that is, chat-oriented dialog systems (CODS) and task-oriented dialog systems TODS. Chat-oriented dialog systems are designed for entertainment purposes, such as in the works of Banchs et.al. [1], and Sugiyama et.al [2]. Task-oriented dialog systems are designed to help the user accomplish specific tasks such as making flight and or restaurant reservations or systems that provide information about specific topics, such as in the works of Kim et.al [3] , and Kubota et.al [4]. While there exist similarities between CODS and TODS, a notable contrast between the two classes cannot be underestimated. This study contrasted the two classes based on two perspectives, that is, complexity and domain of application. With respect to complexity, CODS are quite easy to build and maintain. And as such, they appear too simplistic for applications that do more than answering frequently asked questions. On the other hand, TODS are harder to develop and maintain. They can deal with less variety in terms of user input but are capable of handling and generating all kinds of linguistic phenomena such as grounding and information revision. With respect to domain of application, CODS are generally applied as open domain systems whereas TODS are largely closed domain systems.

The success TODS have had in some domains did have some consequences. One, there is increased reliance on such systems by humans worldwide as argued by Kurzweil [5] &

Lopez [6] . And two, Other new domains have joined in to demand the services of TODS, as highlighted in AI magazine [7]. All these consequences are being propelled by the recent advancements in conversational technology. One important reality is that despite the advancement in technology, there exist domains where TODS are not serving or TODS provide very basic services. A primary reason is that the domain of application dictates the functionality of such TODS. The problem presented here is that some domains have complex conversational requirements that are hard to capture or have not been embedded in such TODS. Examples of domains with complex requirements include but are not limited to health and telecommunication. While an example of a persistent but complex conversational requirement is the ability to advance a conversation logically. The challenge is that these complex requirements are prerequisites for these new domains to flourish. In that context, the argument fronted by the study was that for TODS to serve in the mentioned domains, they must be equipped with these complex requirements. The study notes that the example given in the study is not the only requirement that is needed for modern TODS. This has been discussed in the subsequent sections.

A notable fact identified within the study was that, the behaviour of a TODS is dictated by its domain of application. The study argued that any effort to develop TODS to serve in new domains should commence with understanding the domain with respect to human conversations within such domains. Therefore, to detail and comprehend these diverse and conflicting domains specific needs that would arise during a conversation, the study had to characterize the human conversation in dialog systems. But first, having factored several definitions for what made or constituted a conversation, a working definition for a conversation was adopted as;
"A progression of exchanges among participants, where each participant is indeed a system that changes internally as a consequence of experience", as discussed by Dubberly et.al [8].

To shed-light-on the problem, the study considered the behaviour in some scenarios. The scenarios of interest are more pronounced in domains such as negotiation, bargaining, troubleshooting and diagnosis.
First, consider some human behaviour that seems impossible to ignore or eliminate. In this context, consider human tendencies such as the tendency to interrogate, the tendency to

negotiate, tendency to diagnose or the tendency to justify. These are some examples of natural inherent human behaviour which is difficult to eradicate. In reference to the domains mentioned, there are some tasks where such tendencies are required. The tasks include but are not limited to the following; - Diagnosis context -where "*a doctor cross-examining his patient*". The purpose of cross-examination is to get more information during a diagnosis. Buying context -where a buyer may negotiate pricing with a seller; and Teaching context - where a teacher may justify or be asked to justify something during teaching.

Paying attention to human–machine conversation, using a text-text interface the doctor's context is described in the following way. A user interacts with a machine (doctor) to seek medical advice. The user may ask questions by text and the machine display the results by text on some output screen. However, some of the user questions may prompt the machine to ask questions for the user to respond. This process may continue until a time that the machine feels it has sufficient information to conclude or advise. The user may ask another question related to the earlier question expecting a more specific response. The purpose of providing more information is to aid in getting a more specific response. In other cases, mentioned by the study it could mean to influence a change of decision-like in the case of bargaining or to seek clarity-like in the case of tutoring. In general, such a progression of exchanges may help a user narrow down a search in a more natural way that ordinary searches do not facilitate.

Tasks in diagnosis, negotiation for instance, naturally demand conversations to progress logically. In the study, this has been defined as advancing the conversation. Of course, the advancing need to be logical if there is a goal to be achieved. In the context of a domain that requires diagnosis; a TODS being goal centric, must demand the occurrence of such advancing conversation naturally. Both diagnosis and negotiation scenarios presented are achievable in human-to-human conversations and can be done with reasonable success in human-to-machine conversation. Therefore, the study emphasized that if TODS were to serve domains with such, needs, then it is mandatory for such behaviour to be modelled in dialog systems. A setback here is that, human conversation is considered an artificial intelligence (AI)- hard problem, as argued in the works of Dubberly et.al. [8]. That is, something that could not be done without first solving all the problems of AI. However, they expressed optimism in that it is possible to break down different aspects of the human conversation so that it could be solved piece by piece. The implication here was that while it was difficult to address all possible patterns of a conversation solving piece by piece was a possibility. In this context, the study noted that there are many other aspects of the human conversation which could enhance TODS capabilities. However, advancing the conversation was just one aspect of interest that was deemed important with regard to task achievement. The other aspects were out of the scope of the study.

A fundamental argument presented in the study is that some domains demand certain conversational aspects. Therefore, if developers were to make dialog systems that emulate humans in these domains, and for these domains to flourish, then the developers, need to possess a better understanding of the human conversation. Of course, if they were to make such systems right. The study deems it prudent for developers to consider first, the setting of the conversation, and second, the model of the conversation as discussed in the works of Boyd [9] and Pask [10]. Then adopt the appropriate model of the conversation to a particular setting, during the design of digital systems as argued by Mugoye et.al in [11]. To give a better insight into the human conversation, the study interrogated the works of Dubberly et.al [8] , Boyd [9] and Pask [10]. Dubberly et.al, in their work, provided insights on the complexities in human conversation, whereas Boyd and Pask focused on the essential characteristics, the settings and the various models of the conversation. While these researchers and many others seem to differ in characteristics and settings for a conversation, they did however agree that the presence of "peer to peer exchanges", also known as the progression of exchanges, is one resilient and essential characteristic that persisted across all settings for conversation. In their view, this progression needed to be within a specific domain.

To shed-light-on the problem, the study considered the behaviour in some scenarios. The scenarios of interest are more pronounced in domains such as negotiation, bargaining, troubleshooting and diagnosis.

First, consider some human behaviour that seems impossible to ignore or eliminate. In this context, consider human tendencies such as the tendency to interrogate, the tendency to negotiate, tendency to diagnose or the tendency to justify. These are some examples of natural inherent human behaviour which is difficult to eradicate. In reference to the domains mentioned, there are some tasks where such tendencies are required. The tasks include but are not limited to the following; - Diagnosis context -where "*a doctor cross-examining his patient*". The purpose of cross-examination is to get more information during a diagnosis. Buying context -where a buyer may negotiate pricing with a seller; and Teaching context -where a teacher may justify or be asked to justify something during teaching.

Paying attention to human–machine conversation, using a text-text interface the doctor's context is described in the following way. A user interacts with a machine (doctor) to seek medical advice. The user may ask questions by text and the machine display the results by text on some output screen. However, some of the user questions may prompt the machine to ask questions for the user to respond. This process may continue until a time that the machine feels it has sufficient information to conclude or advise. The user may ask another question related to the earlier question expecting a more specific response. The purpose of providing more information is to aid in getting a more specific response. In other cases, mentioned by the study it could mean to influence a change of decision-like in the case of bargaining or to seek clarity-like in the case of tutoring. In general, such a progression of exchanges may help a user narrow down a search in a more natural way that ordinary searches do not facilitate.

Tasks in diagnosis, negotiation for instance, naturally demand conversations to progress logically. In the study, this has been defined as advancing the conversation. Of course, the advancing need to be logical if there is a goal to be achieved. In the context of a domain that requires diagnosis; a TODS being goal centric, must demand the occurrence of such advancing conversation naturally. Both diagnosis and negotiation scenarios presented are achievable in human-to-human conversations and can be done with reasonable success in human-to-machine conversation. Therefore, the study emphasized that if TODS were to serve domains with such, needs, then it is mandatory for such behaviour to be modelled in dialog systems. A setback here is that, human conversation is considered an artificial intelligence (AI)- hard problem, as argued in the works of Dubberly et.al. [8]. That is, something that could not be done without first solving all the problems of AI. However, they expressed optimism in that it is possible to break down different aspects of the human conversation so that it could be solved piece by piece. The implication here was that while it was difficult to address all possible patterns of a conversation solving piece by piece was a possibility. In this context, the study noted that there are many other aspects of the human conversation which could enhance TODS capabilities. However, advancing the conversation was just one aspect of interest that was deemed important with regard to task achievement. The other aspects were out of the scope of the study.

Aspect of interest	Questions that need to be answered as a way to pursue the solution
Primary Approach / category Primary approaches for classifying dialog systems.	. What is the salient description of the approach?
View on dialogue flow Approach representation of flow of dialogue.	. Is the approach's view or representation of the flow of dialogue ideal for advancing conversations?
Current Capability Current capability of the approach.	. Is the approach sufficient to model advancing conversation?
Potential Capability Potential capability of the approach.	. Can the approached be stretched to model advancing conversation?
Implementation Implementation type.	. How else can the approach be implemented?
Implementation challenge Challenge in Implementation.	. What is the challenge in implementing a hybrid? Is the hybrid a reliable method for guaranteeing advancing conversation?
Dependency Dependency on external capabilities.	. Does the approach rely on hand crafted techniques?
Dependency Shortcomings Dependency problems.	. What would be the problem of dependency on hand

Figure 1.1. The lenses for interrogating common architectures.

The following architectures were subjected through the lenses as shown in figure 1.1. The architectures were based on broad classification according to the implementation approach. They were; - State-Based, Frame-Based, Plan-Based and Agent-Based. The limitations inherent in each approach informed the choice of solving the problem by defining a novel architecture. Figure 1.2 compares architecture capabilities. The approaches have been further discussed in section 2.

Aspect	Finite-state based Approach	Frame-based Approach	Plan-based Approach	Agent-based Approach
Description	Represents a dialogue as a sequence of pre-determined states.	Views a dialogue as a series of questions to a user. The system uses the responses to fill slots in a template in order to perform a task.	Utilize plan-based modelling of dialogue. Breaks down the overall task into smaller goals and plans, and controlling the interaction to accomplish them.	View dialogue as an interaction between two agents. Each agent is capable of reasoning about its actions and beliefs
View on dialogue flow	The flow of dialogue is specified as a set of dialogue states with transitions denoting various alternative paths through the dialogue graph.	The flow of the dialogue is determined by user input. Dialog system asks for values for the slots by using predefined actions for each set of known slots.	Plans are frames, plans cannot expand during runtime. The flow of the dialogue is similar to frame based.	The flow of the dialogue is determined by the rules instilled in the agent and how it perceives its environment. Considered more likely to address the limitations in the other approaches.
Capability to model advancing conversation	Absent, it compels users to follow the pre- defined paths for the different states.	Absent, its hard / imposible to specify posible responses upfront	Absent, its hard / imposible to specify posible responses upfront	Present, It has been advocated to offer the dialogue that evolves dynamically based on the current context.
Implementation type	FSM	Frames / Slot filling	Plans models	Single / Multi-agent
Implementation challenges	Complicate the architecture Affects portability	Complicate the architecture Affects portability	Complicate the architecture Affects portability	Single / Multi-agent
Require support of hand-crafted rules	YES	YES	YES	NO
Deployed example available?	YES	YES	YES	Single Agent -YES Multi-Agent -NO DATA

Figure 1.2. A comparison of architecture capabilities.

Although building a dialog system that can address all possible patterns that may exist in a conversation and complete dialogue with a human is still challenging. The lenses explored applied to one aspect of interestingness in the study. What appears to be missing is a reliable, portable technique and architecture to promise or guarantee advancing conversation. In that context, the study applied a novel method and technique to deliver the ability to guarantee logically advancing conversations in a dialog system. Assuring the desired progressive conversation in a dialog system was achieved by exploiting reinforcement learning on a multi-agent system. Here multiple intelligent agents collaborate towards tracing the structure of exchanges, tracking the context, and utilization of a control mechanism. Discourse familiarization and action selection are supported through reinforcement learning.

With respect to an advancing conversation, the completion of a diagnosis task successfully and giving a valid recommendation is considered a PASS. Otherwise, it's a Fail. A solution that guarantees to advance a conversation offer a number of gains; one, it gives a user a chance to present more information gradually thereby eliminating the need to remember everything at once. Two, it makes the interaction enjoyable. Furthermore, a logical progression of exchanges especially during a search will relieve users of systems the burden of too much-thinking upfront during an interaction. This is ideal in domains where the user(s) may be expected to be under duress. Another important gain is to enable TODS to serve in new domains, which are characterized by complex requirements such as bargaining or negotiating, intelligent information retrieval or advisor systems, intelligent tutoring, and advanced navigation systems. While this type of exchange may help eliminate user frustrations when interacting with such new interfaces, one notable drawback is that such an interface may prolong the process of enquiry. That, however, is a small price to pay considering the consequences related to the use of frustrating human-machine interfaces. Dubberly et.al. [8] and Mugoye et.al [11] describe the consequences related to the use of frustrating human-machine interfaces to include negative feelings, underutilization and avoidance of systems.

The study pointed out that sentence structure analysis, the ability to refer to context a promising policy selection technique were necessary ingredients for achieving logical advancing conversations. Besides, the presence of a critic "*function*" makes it even better as the use continues. The approach in the study emulates the human conversation model where the participants maintain focus on the context by continuously tracking the state of the exchanges while internalizing the relationship as the conversation moves forward. The study views the progress to satisfy the user's goals adequately in a conversation as the progress when the user not only complete a task but also enjoys the interaction.

1.2 Problem Statement

Conversational artificial intelligence, also known as conversational AI, has attracted interest owing to the fact that task-oriented dialog systems have gained traction on industrial use and that their applications are being extended to new domains, as emphasized in Kurzweil [5], Lopez [6] and AI magazine [7]. However, this extension has introduced two new problems; First, it creates increasing demands for a more natural interaction to fit well in a domain. Second, some new domains demand more complex types of conversations. Example of such include conversation with the aspect of progressive peer-to-peer exchanges or advancing conversations, as discussed in section 1.1. As a consequence, very few commercial TODS are serving in some domains, and for some domains they are just providing basic services, as suggested by Pieraccini [44].

The argument fronted by the study is that domains that demand behaviour like negotiation, bargaining, diagnosis, and troubleshooting require a complex types of conversation. In this context, such conversation requires the capability to advance logically, for a goal to be achieved. Therefore, for TODS to serve in such domains the ability to assure logically advancing conversation, without losing context is mandatory. This problem becomes more aggravated in cases where such a task-oriented dialog system is supposed to offer advice. The study sought to fill the gap by providing dialog system with the capability to offer advancing conversation, with reference to the context of the conversation. The study proposed a new architecture and method as a way to guarantee advancing conversation, while enabling portability to other domains. The anticipated gains will be that TODS will serve these new domains, TODS will have simplified architectures easy to study and their application areas will increase.

1.3 General Objective

The objective of the study was to develop a Multi-agent based architecture to guarantee advancing conversations in TODS.

1.3.1 Specific Objectives

To achieve the general objective, this research will involve the following specific objectives. The specific objectives must be achieved; -

1) To formulate the architecture of an agent based dialog management.
2) To implement the agent based architecture in a dialog system prototype.
3) To train the dialog system on initial domain specific data.
4) To evaluate the performances of instances of dialog system prototype in specific domain.

1.4 Research Question

In domains where bargaining or negotiation are mandatory and within the context of dialog systems, can a new architecture and method be applied to support conversation that advance logically?

1.5 Scope and significance of the study

The scope of this study focused on text-to-text TODS, with emphasis on logic and progression of the conversational aspect. Robustness, error recovery, quality of knowledge source or traditional language data corpus were deemed out of scope.

Significance of the study

To the researchers, the results from the study can be a learning paradigm for other researchers in conversational software. To users, by eliminating the need for upfront restructuring of sentences users benefit from a more natural conversation and thus enjoyable user experience. To developers, it is expected that dialog systems adopting this approach will have better conversational capabilities, which could not be guaranteed in the other approaches. To the industry, it is expected to improve task-oriented dialog systems commercial exploitability and application. Generally, it is a great stride in the quest of making machines more of human conversational partners. And it will translate to increase on the usability of useful applications or machines.

1.6 Assumption and Limitation of the study

Assumptions

The implemented dialog system did run on mobile and other platforms with little or no modification. Conversation with the dialog system will always be in English. A user's goal remains to accomplish a task. Gender and or age of a user did not affect the functionality of the prototype. The prototype was not designed for use in circumstances where underlying medical conditions prevail.

Limitations

The research remained cognizant of issues in MAS that needed to be addressed; such issues include but are not limited to emergent behaviour. While such issues were unavoidable in the study and design of complex systems, they fell out of the scope of the study. Error recovery and robustness were important abilities for dialog systems however, they fell out of the scope of the study. In addition, the study did not try to solve issues that deal with the completeness and availability of corpus datasets.

1.7 Thesis organization

The thesis is organized in five chapters with each handling specific aspects of the study. Chapter one provides the background of the study, the problem statement, research objectives, contribution, scope and significance of the study. Assumptions and limitations of the study concludes the chapter. Chapter two provides in-depth review of related work. This chapter discusses and critiques the major types of architecture, used for dialog systems, and a summary of the review closes the chapter. Chapter three provides the methodology employed in the undertaking of this research. Discussions on the design and development of the prototype has been provided here. Chapter four presents and discusses results from the study. Chapter five presents the contributions and conclusion of the study. The chapter ends with a recommendation for further study.

CHAPTER TWO

LITERATURE REVIEW

INTRODUCTION

This chapter introduces dialog management in section 2.1 and provides a generic architecture for dialog management in 2.2. Then presents the critical sections or functionality that separate dialog systems in 2.3 through 2.6. The sections further reviews and critique common approaches to dialog systems, that is, the finite state-based, the frame-based, the plan-based, and agent-based dialog system respectively. Section 2.7 presents a synopsis of related works that were considered significant to the study. Section 2.8 suggests what informed the approach taken in the study. The state of the art dialog systems is presented in 2.9. Section 2.10 presents a synopsis of machine learning while 2.11 discusses the evaluation of the architecture. Finally, a summary of the review concludes the chapter in section 2.12.

The focus of dialogue management is to find a dialog system's best response given user's interaction history. Precisely, dialogue management controls the whole dialogue process, and its design directly affects the performance, capability and classification of the dialogue system. Dialogue management mainly includes two tasks namely; dialogue state tracking and dialogue policy learning. Dialogue state tracking determines the current user target based on multiple rounds of conversations between the system and the user. It also provides the basis for later decisions. Dialogue policy selection, on the other hand, selects an executable action based on the results of the dialogue state tracking. There exist various approaches to dialogue management. The approach determines the classification of a dialog systems architecture. For instance, the state-based mechanisms employ state-based architecture. There also exists a wide collection of methods on how to implement a dialogue management mechanism. As such, we have pure or basic, extended or supported, and hybrids. The distinction lies in the implementation; generally, pure implementations employ the architecture without external support. The extended implementations employ the architecture with external support. And the hybrid implementations combine more than one architecture. The study noted that any dialog system is supported by one of the architectures. In that context, the approaches range from finite state machines and frame structures through intelligent agents.

The next sections present common approaches to dialog systems and review some of the common dialog systems that have had success in the various areas of application.

2.1 Dialog manager

The dialog manager is a component that manages the state of the dialog, and dialog strategy. The architecture of the dialog manager impacts the resultant dialog system. McTear [17], Barnard et al. [18] suggested that the limitations of a given architecture impact the capabilities of the resultant dialog system.

It is from this premise that the study suggested MAS_DM architecture, with the view of enhancing dialog management, to be able to support advancing conversation within a closed domain. The rationale of MAS_DM architecture is to avoid the architectural limitations discussed and over reliance on handcraft techniques.

2.2 Traditional architecture for dialog manager

The traditional architecture utilizes a generic model for dialog management. That is, the dialog manager is composed of a generic dialog management mechanism, with access to specific dialog management module. This module may have a direct access to a knowledge base, Liu [19]. There are two calibrations, one where the dialog manager has access to a knowledge source, as presented in figure 2.1, while the other is where the dialog manager has access to a task ontology, as presented in figure 2.2.

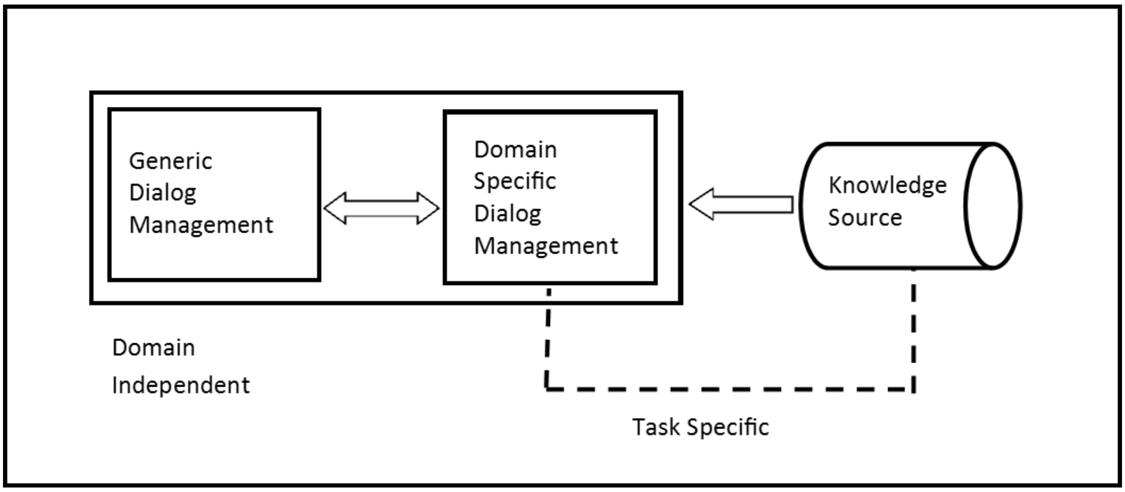


Figure 2.1. A generic dialog manager, for domain independent dialog management based on traditional approach. Adopted from Liu [19].

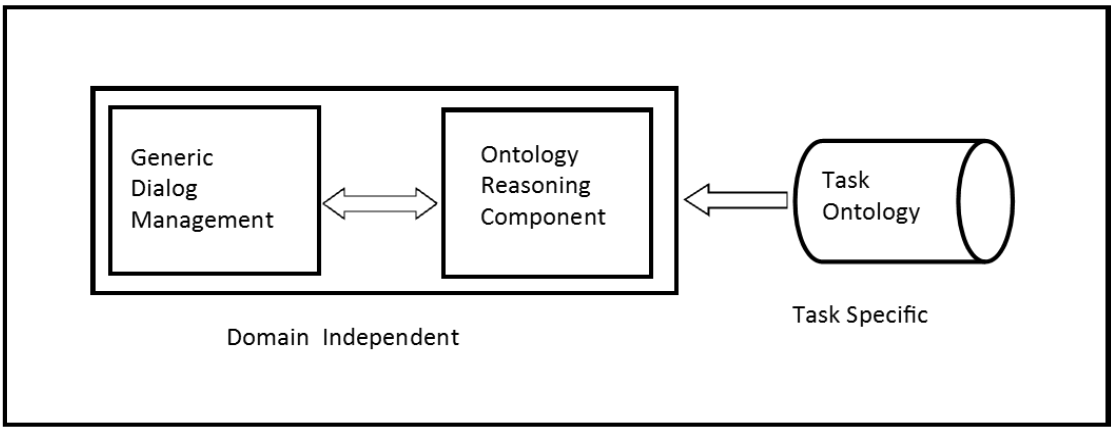


Figure 2.2. An ontology based generic dialog manager, for domain independent dialog management based on traditional approach, adopted from Liu [19].

The generic models have worked well for smaller simple problems, however, for complex problems, the models suffer the drawbacks discussed in section 2.1 and 2.2 of the literature. In an attempt to counter the drawbacks mentioned, the generic dialog manager models often require the support of different handcrafts to introduce a missing capability or anticipated behaviour. Such handcrafted techniques have had success in numerous circumstances, especially for simple problems. As a result, the handcrafts have gained popularity as the remedy for fixing missing functionalities. However, handcrafts introduced the problems described in subsequent sections of the literature.

2.3 Finite-state based dialog systems

A finite-state dialog system , as in Guan et al. [20] adopts a finite-state architecture. A high-level breakdown of this architecture identifies three major components namely; dialogue state, state transition and dialogue policy. A dialogue state represents the state of the dialogue in every moment. The dialogue state depends on a state transition function which is responsible for updating the dialogue state taking into account the user and system acts. The dialogue policy is in charge of deciding which transition function applies to a state between a group of transition functions. Figure 2.3 shows the high-level description of this architecture graphically.

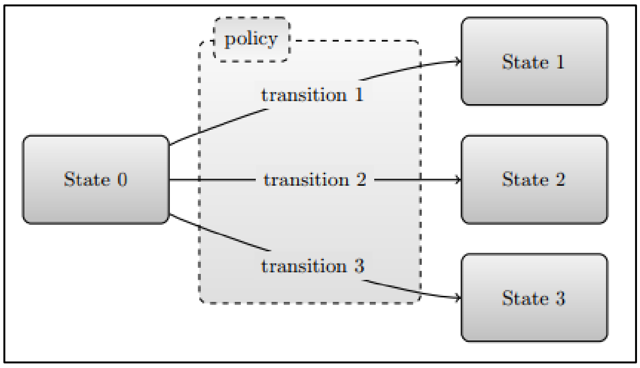


Figure 2.3. Simplified FSM architecture example.

This architecture represents a dialogue as a sequence of pre-determined states, as argued in Shi et al. [21]. The flow of dialogue is specified as a set of dialogue states with transitions denoting various alternative paths through the dialogue graph. The system takes control of the dialogue, by producing prompts at each dialogue state. The system recognizes specific words and phrases from user's input and produces actions based on the recognized response. A primary characteristic of this approach is that a user's input is usually restricted to single words or phrases which provide responses to carefully designed system prompts, as described by McTear [22]. The finite-state approach appears to be the most common, it bases its strength on its simplicity. It is commonly preferred in simple, well-structured and routine tasks. Successful dialog systems developed using this approach include but not limited to Bennett et al. voiceXML application [23].

2.3.1 Critique of Finite-state based Approach

Although simplicity inherent in the approach may serve as its strength, it also acts as a fundamental bottleneck, as such, it cannot handle complex dialogs. Sutton et al. [24] view the finite-state approach as not suitable to manage complex dialogues due to the lack of flexibility. In essence, it compels users to follow the paths defined for the different states. Another drawback is that it cannot allow the user to take the initiative of the conversation.

The finite-state approach to realizing dialog systems is not suitable for advancing conversations for three reasons. First, due to its simplistic nature, it is not possible to model patterns essential to advancing a conversation such as negotiation. Second, the rigidly predefined dialog paths become a bottleneck to advancing conversations. Third, its inability to allow the user to take the initiative in a conversation make it a poor choice. For these reasons most currently available commercial systems that rely on this approach requires the support of handcrafted rules to improve their performance.

Some limitations inherent in the finite-state approach, for instance, simplicity, and rigidness may be addressed through the use of handcrafted techniques. However, handcrafts present new challenges. First, they make the architecture complex, second, handcrafts cannot be ported.

2.4 Frame-based Dialog systems

The Frame-Based dialog system also known as Form-Filling [25], adopts a Frame-Based architecture. This architecture in its basic form uses the concept of frames. A frame is a data structure consisting of a set of slots, concepts the user can talk about, which can take on predefined values. The dialog system asks the user for values for the slots by using predefined actions for each set of known slots. The system takes control of the dialogue, by producing prompts at each dialogue state. The system recognizes specific words and phrases from the user's input and produces actions based on the recognized response. Figure 2.4. presents an example of Frame-Based architecture.

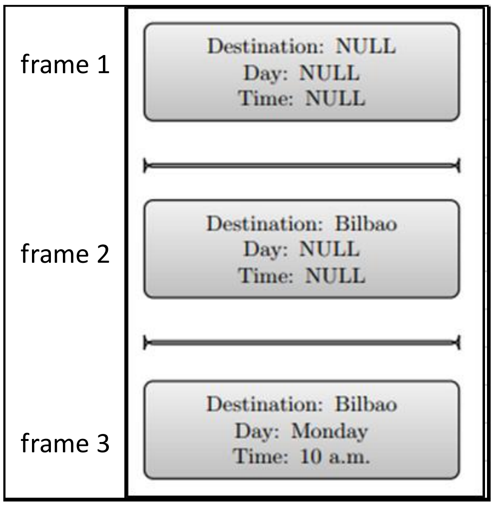


Figure 2.4. Simplified frame based architecture example.

Based on this architecture, a dialogue is viewed as a series of questions to a user. The system utilizes the responses to fill slots in a template to perform a task. The user is specifically asked questions that enable the system to fill slots in some template to perform a task. In this context, the dialogue flow is not pre-determined but depends on the content of the user's input and the information that the system has to elicit. The frame-based architecture guarantees a more flexible dialog system than the finite-state based approach, as the dialogue flow is event-driven, not predetermined. It provides room for multiple slots filling which enables the system to process the user's over-informative answers and corrections. This may guarantee reduced transition time of the dialogue resulting in a more natural and efficient dialogue flow. As such, examples of successful dialog systems developed using this approach include but not limited to the Philips automatic train timetable information system presented by Steinbiss et al. [25] and the speech-interactive automation system, by Zeigler et al. [26].

To overcome simple dialog problems, the basic frame architectures are commonly extended. This extension may include additional data structures such as a control table and other hand-crafted rules to decide the operations to perform based on the content of the frame. Examples of dialog systems that use an additional control table are presented in the works of Zue et. al. [26] and Seneff et. al. [27].

Within the specifics of the transfer functions and policies, it can be considered that the dialog managers state is represented by the frame and the transfer function by the updating process of the frame. That is the filling of the slots. Regarding the policy, almost every implementation of a dialog manager has its customized policy implementation.

2.4.1 Critique of Frame-based Approach

The frame-based approach is perceived to be more flexible in comparison to the finite-state based approach. While this may serve as its strength over the finite-state based, Frame-based approach relies heavily on some form of natural language input to, permit the user to respond more flexibly to the system's prompts and to correct errors of recognition. A major limitation of the frame-based approach is its unsuitability for modelling complex transactions. This limitation is due to the fact that different users have different levels of knowledge and state of the world can change during the conversation. As a consequence, the range of responses needed is wide and it's impossible to specify it in advance. Furthermore, from a developer's point of view, it is very challenging to predict which rule is likely to come true in a particular context. A considerable amount of experimentation may be required to ensure that the system works as expected.

With regard to achieving advancing conversations, the frame-based approach looks like a better choice in comparison to finite state. This observation can be attributed to the fact that the approach provides the capability to model patterns essential to advancing a conversation such as negotiation. However, the frame-based approach presents three major limitations, first, this approach requires a very wide range of responses to be specified upfront. It is impossible to specify such in advance. Second, the approach gives a very diminutive guarantee that an appropriate rule will be fired at a particular context. Consequently, there is no guarantee that a conversation may advance logically. Finally, dialog management based on this approach cannot be optimized.

The frame-based approach can be extended using handcraft to improve performance. However, some limitations inherent in the frame-based approach, for instance, the need to specify responses upfront cannot be resolved by a handcraft. Besides, a limitation such as having no guarantee that an appropriate rule will be triggered falls contrary to advancing a conversation. The chance of such a limitation to be resolved by a handcraft are slim.

2.5 Plan-based Approach

Plan based approach employs a plan based architecture. Among other researchers, McTear, Perrault et al. and Grosz and Sidner describe the primary underlying concept behind plan-based approaches. McTear [22] ,Perrault et al. [27] and Grosz and Sidner [28] describe plan based approach as an approach that treats each user utterance as though it is an action performed in order to reach some goal. Skantze [29] adds that the dialogue actions link the current state of the conversation to the achievement of their goal. Plan-based dialog systems utilize plan-based modelling of dialogue, which involves breaking down the overall task into smaller goals and plans, and controlling the interaction to accomplish them, Wu et al. [30]. Such a system identifies the overall goal a user wishes to achieve then the system develops a plan, composed of a series of dialogue actions.

Research portrays the existence of an overlap between plan-based and frame-based approaches. Frame-based systems are perceived as one way of achieving plan-based dialogue. Frames structured in a hierarchy, for example, are naturally amenable to plans which can be de-constructed into lower-level goals. Therefore, a plan based architecture is a variation of frame based architecture. For instance, RavenClaw [35] and Topic Forest [30], both match users' utterances or the system state to a particular frame in a tree and uses this to infer the user's goals. Similarly, Topic Forest by Wu et al. [30] has a plan-based dialogue management structure that utilizes hierarchical relationships, that is, topic trees to represent the information items required of different domain topics. Examples of systems that claim to have implemented a plan-based approach are described in the works of Chu-Carroll and Carberry [31] and Moore & Paris [32].

Wang [33] describes plan-based methods to have the ability to provide scalable solutions to dialogue management, containing the required intelligence to automatically decide the pathways through a conversation. However, critics claim that plan based approaches do not actively develop and expand upon plans during runtime, as in the works of Laranjo et al. [34] .

2.5.1 Critique of plan-based Approach

Laranjo et al [34] argue that the plan based approaches do not accommodate development and expansion of plans during runtime. Besides, plan based approach being an implementation of the frame-based approach suffers all the drawbacks of its parent architecture, i.e. frame-based, discussed in section 2.2.

2.6 Agents-based Approach

Agent-based dialog systems view dialogue as an 'interaction between two agents, each of which is capable of reasoning about its actions and beliefs McTear [22]. Agent-based approach is considered more likely to address the limitations in the other approaches. It has been advocated to offer the dialogue that evolves dynamically based on the current context, Nguyen et al. [35]. In most cases agent based refer to single agent systems, single agent systems have the potential to get overwhelmed as the data corpus increases, and the structure of the conversation get more complex.

A rationale for the use of multi-agents in the dialog systems has been the recognition that certain problem-solving tasks involve a cooperative effort between individuals, as in Niazi et al. [36], especially when agents have differing capabilities. The embodiment of agents, including their reasoning and 'intelligence', is also fundamental. An action or response in an agent-based DS is the outcome of the combined contributions of each relevant agent which have engaged in a collaborative activity based upon the rules of engagement that have been instilled within them, as described in Wooldridge [37]. Blaylock [38] presents TRIPS as an example of an intelligent dialogue agent. In such a system, the dialogue evolves dynamically based on the current context. Other examples are CMU's RavenClaw [35] and DISCO [36].

2.6.1 Critique of agent-based Approach

One primary drawback of this approach is that systems are usually hard to build, and the agents themselves are very complex.

2.7 Significant Related Work

Different solutions to address various dialog system shortcomings that emanate from architectural models have been attempted. For instance, Litman et.al [39], employed the finite state model for the dialog strategy. In their work, they discussed how to reconstruct error in speech, during a dialogue. Their focus was on how the system adapts to errors in speech. Marjan Ghazvininejad et al. [40], ventured a study on a conversational question answer system. In their study, they presented a fully data-driven neural conversation model that effectively exploits external knowledge, without explicit slot filling. Their aim was producing a more contentful responses. They did not, however, use the agency or even consider the structure of conversation in their solution.

Solutions aimed at enhancing the conversation have also been attempted. For instance, Trieu et al. [12] and Ghazvininejad et al. [40] have in their studies discussed the conversation. However, they pursued different goals. In their work, Trieu et al. [12] the goal was to prolong a conversation with the view of entertaining a user, whereas the goal of Monroe et al. [41] was to achieve responses that would have more content. Trieu et al. [12] discussed the conversation from the view point of being entertaining. They explored the game refinement theory to develop a dialogue system with entertaining conversations. They suggested a method to improve the current goal-driven dialogue systems which support users for specific tasks while satisfying users' goals with entertaining conversations. Their emphasis however, was to generate entertaining conversations by reasonably prolonging the original too short dialogue. In their work, they explored a different pattern of a conversation to prolong a conversation that otherwise they would view as short. They did not, however, venture into reaffirming whether the prolonged conversation could advance logically.

Solutions applying reinforcement learning have been attempted. Gellert Weisz, et al. [13] have explored deep reinforcement learning to address policy optimization in dialog systems. They explored reinforcement learning to find a policy describing how to respond to humans, in the form of a function taking the current state of the dialogue and returning the response of the system. The viability of reinforcement learning in dialog systems have further been demonstrated by the works of Singh et al. [42] and Li et al. [41]. The authors here, in their work did not consider architecture as an impeding factor.

Various solution intended to enhance dialog systems have been attempted. Stolke [14] in his work, focused on grammars recognition and suggested a technique for addressing it. Stoyanchev [15], in his work, focused on the ability to understand errors and recover from them more quickly from speech. While Sun [16], focused on how to understand out-of-vocabulary words from users. Stolke, Stoyanchev and Sun did not however address these problems from the architecture point of view, they did however use generic models, supported by various handcrafts.

Other solution in addressing dialog systems challenges that dealt with other human issues included, Walker [43]. Walker discusses the system's ability to adapt to features of both the dialog partner such as age, and the modality to apply. Baskar [44], discusses the system's ability to conform to personalized dialogues between a human and a software agent. Gnjatovi´c and Rosner [45], in their work, created a user-adaptive dialog system that identify the emotional state of the user and provide support in cases where the user seems frustrated. Rosne et al.'s intention was to help to solve the Tower-of-Hanoi puzzle. The researchers here, did not consider architecture as an impeding factor. In their work, they explored various handcrafts.

Other interesting works on dialog systems included, Jameson [46], and Johansson [47]. Both researchers work discussed ways of supporting system use and supporting information acquisition. Nothdurft et al [48] created a dialogue which is adaptive to the user knowledge. Their system made assumption over the user knowledge by observing critical events within the dialogue. Based on events extracted from the dialogue, the system generates explanations and selects the appropriate type of explanation so that the user can be expected to be capable of solving the task.

2.7.1 Critique of Related Work

In an attempt to enhance the performances of dialog systems, different solutions have been tried. The study categorized solutions into two categories, namely type one and type two. Type one solutions referred to solutions that necessitated the redesign of the underlying architecture or adoption of a new architecture. For instance, using the frame-based models to overcome some limitations of the finite-state models. Type two solutions referred to solutions that involved the use of handcrafted techniques to overcome architectural limitations. Both type one and type two solutions have yielded success in several areas of applications. However, both solutions introduced new challenges that need to be addressed. By the time of the study, most enhancements on dialog systems were either supported by the underlying architecture or the adopted handcraft technique. It is significant to mention that the paradigms explored were not feasible for complex enhancements. A possible example of a complex enhancement is a case where a task-oriented dialog system was designed to offer advice. The study argued that such a dialog system required to have the ability to advance a conversation, the ability to reference items mentioned earlier in a dialogue and also keep track of different sub-dialogues, within the context of the dialog. Of course, it should be cognizant of the topic and the main goal of dialogue is to be able to make appropriate moves.

In the works reviewed, the approaches worked well for simple enhancements and the success or failure, depended on how the architecture did support the dialog management. With regard to solutions that focused on the conversation as a problem, Trieu et al. managed to prolong a dialog systems conversation to entertain a user. However, Trieu et al. [23] did not reaffirm whether prolonging the conversation required the conversation to advance logically. These considerations informed and justified the need to redesign the underlying architecture. The study's position was that a different paradigm needed to be considered if complex enhancements were to be supported, the architecture could be ported to other domains. Finally, if there was the need to guarantee conversations that advance logically.

2.8 Criteria for arriving at agent based solution

The common architectures for building dialog systems were interrogated, based on six aspects of interest to the study. First, a brief description was provided, then securitized based on the following aspects; - view on dialogue flow; capability to model advancing conversation; Implementation type; Implementation challenges; Need for external support (hand-crafted rules); and a deployed example if any. Table 2.2 provides a summary of the criteria applied in scrutinizing the architectures.

The drawbacks imposed by the approaches mentioned in the study bring to light the need for alternatives for dialog management approaches that are more reliable and that do not significantly complicate the architecture.

Aspect	*Finite-state based Approach*	*Frame-based Approach*	*Plan-based Approach*	*Agent-based Approach*
Description	Represents a dialogue as a sequence of pre-determined states.	Views a dialogue as a series of questions to a user. The system uses the responses to fill slots in a template in order to perform a task.	Utilize plan-based modelling of dialogue. Breaks down the overall task into smaller goals and plans, and controlling the interaction to accomplish them.	View dialogue as an interaction between two agents. Each agent is capable of reasoning about its actions and beliefs
View on dialogue flow	The flow of dialogue is specified as a set of dialogue states with transitions denoting various alternative paths through the dialogue graph.	The flow of the dialogue is determined by user input. Dialog system asks for values for the slots by using predefined actions for each set of known slots.	Plans are frames, plans cannot expand during runtime. The flow of the dialogue is similar to frame based.	The flow of the dialogue is determined by the rules instilled in the agent and how it perceives its environment. Considered more likely to address the limitations in the other approaches.
Capability to model advancing conversation	Absent, it compels users to follow the pre- defined paths for the different states.	Absent, its hard / imposible to specify posible responses upfront	Absent, its hard / imposible to specify posible responses upfront	Present, It has been advocated to offer the dialogue that evolves dynamically based on the current context.
Implementation type	FSM	Frames / Slot filling	Plans models	Single / Multi-agent
Implementation challenges	Complicate the architecture Affects portability	Complicate the architecture Affects portability	Complicate the architecture Affects portability	Single / Multi-agent
Require support of hand-crafted rules	YES	YES	YES	NO
Deployed example available?	YES	YES	YES	Single Agent -YES Multi-Agent -NO DATA

Table 1.1. The criteria for scrutinizing the architectures.

2.9 State of the art dialog systems

The modern state of the art dialog systems are designed for commercial purposes. Such purposes are normally specific. As such, it is fundamental to note that the purpose, cost or budget and application domain play a significant role to determine the architecture in use. In that context, designers impose severe limitations on the scope of the applications, as suggested by [34], which require a great amount of manual work for the designers. Besides, the architecture or handcrafts used are intended to address a specified reason for building the dialog system. This is so because different capabilities are tied to different architectures. Example of specific purpose include vision-based facial expression recognition and Lidar-based distance detection.

Generally, modern dialog systems use new architectures such as partially observable Markov decision processes (POMDPs) based, or hybrid architectures with support from handcraft techniques. However, POMDP based dialog systems suffer from the fact that only a relatively small number of domain variables are allowed in the model, as in the works of Lu et al [33]. In that context, the new architecture has its set of problems while hybrids are not excepted from the problems discussed in the literature.

State of the art dialog system do exist, however, there is little data in the public domain about them. Mitsuku is a perfect example, Mitsuku chatbot [49] won the Loebner Prize for the most human-like conversational AI for the year 2018. According to its developer Steve Worswick, Mitsuku uses the extended architecture, that is the generic architecture extended with handcrafted rules. Of course, it also suffers the limitations inherent in the architecture. However, since its purpose was not to achieve tasks it is not mandatory that its conversation must advance or depict logic in such advancement if any. Note, the purpose of presenting Mitsuku was not to achieve a one on one comparison but to present a baseline for understanding advancing conversation. For any TODS such advancements need to show logic within the progression.

2.10 Reinforcement Learning

Machine learning is desired because apart from enabling the agent to be autonomous, it provides an assurance that correct and hence purposeful conversation will be attained for future conversations.

A dialogue is a temporal occurrence in a dynamic environment. A dialogue is temporal in the sense that how good an action is; depends on how the dialog progresses further, as in Verena [50]. The environment set-up is dynamic in the sense that perceivable inputs and the state of the environment keep changing. An appropriate machine learning technique needs to be one that can adapt to the features of the environment of interest. In that context, supervised learning as presented in [51] is not capable to handle the dynamism presented by such environments;- it is not possible to present correct input/output move pairs of ideal dialogue strategy behaviour. The complexity inherent in this set-up slightly overwhelms unsupervised learning in that, such algorithms do not have the potential to improve and learn from new situations [51]. Reinforcement Learning (RL) therefore promises better outcome. RL models the problem as a sequential decision process with long-term planning, Verena [50] and Litman et.al [52]. RL learns by exploration and exploitation of current knowledge, as described in Sutton [24] and encourages favorable outputs while discouraging non-favorable outputs. Litman el al. and Sutton further argue that such an ability to explore allows a system to learn strategies that are more robust to unseen and unpredictable states.

In the context of the problem this study attempted to address, RL proves more promising in the sense that RL algorithms improve and learn from new situations. Therefore, the study argued that RL will guarantee better results.

2.11 A synopsis of Advancing conversation

Advancing conversation has been defined as a progression of exchanges that make reference to the previous statement in the current situation, which allows growth within a context. Normally intended to achieve a goal. Advancing conversations are demanded for domains with tasks like negotiation, bargaining, troubleshooting and diagnosis. It is quite an ambiguous aspect to measure in dialog systems since it is subject to human subjectivity and interpretation. However, for TODS the objective of such advancement is to guarantee task achievement. Therefore, it is possible to demonstrate and prove such advancement through task achievement. In the study, task achievement has been relied on heavily as proof of advancement, of course, the advancement must be logical for a task to be successfully achieved. This justifies the need for GQM evaluation, which is highly user-dependent. In this context, the study presents a sample conversation with the objective to determine if the goal was successfully achieved. If yes it can be argued that the conversation advanced logically.

GUS: Do you want a flight leaving at 4.00 pm
CLIENT: Do you have something a little closer to 7?
GUS: Do you want the flight at 7.00 pm?

(a) Interpreting politeness

GUS: Do you want the flight arriving at 8.00 pm?
CLIENT: When does it leave?
GUS: 6.30 pm
CLIENT: How much?
GUS: $25.50 round trip

(b) Some pronominal reference problems

GUS: When would you like to return?
CLIENT: I would like to leave on the following Tuesday, but I have to be back before my first class at 9 am.

(c) Giving a reason for flight preference

FIG. 10. Fragments of real dialogs, with a person simulating the role of GUS.

Figure 2.5. Sample dialog system conversation.

The conversation in figure 2.5 was picked from a TODS named Genial Understander System (GUS). GUS is a flight booking system developed at the Xerox Palo Alto Research Centre, USA. GUS initiates a conversation it proposes a flight whose departure time is 4 pm. The client requests a departure time closer to 7.00 pm. GUS assumes the client wants a flight departing at 7.00 pm. GUS asks if the user is interested in a flight arriving at 8.00 pm. When the client responds by asking the departure time, GUS gives the departure time of 6.00 pm. Subject to human evaluation, first, closer to 7 pm does not necessarily mean 7 pm. Second, there seems to be a problem between the arrival time of 8.00 pm and departure time of 6.30 pm, if this means the same day. The progression depicts a problem with logic in the exchanges.

What can be learnt as argued in the study is, that there is no certainty that exchanges make reference to the previous context. Since logic enables goal realization, it is not certain whether a goal was successfully realized. These exchanges, therefore, fall short of advancing conversation based on the definition. When client ask when it leaves, GUS responds 6.30pm, recall the flight arrive at 8.00pm.

The sample conversation from Mitsuku [49] in is presented in figure 2.6. It depicts progression in the exchanges, however, it can be noted that it pays little attention to the context of the conversation. A reality that cannot be ignored is that CODS are not bounded by defined goals or task to be achieved. In cases where goal or task to be achieved are not mandatory

It becomes ambiguous determine absence or presence of “advancing conversations”; - In other words, the argument as to whether there is advancing conversation can take either side. The study argues that goal or task achievement, is an important measure in understanding advancing conversation.

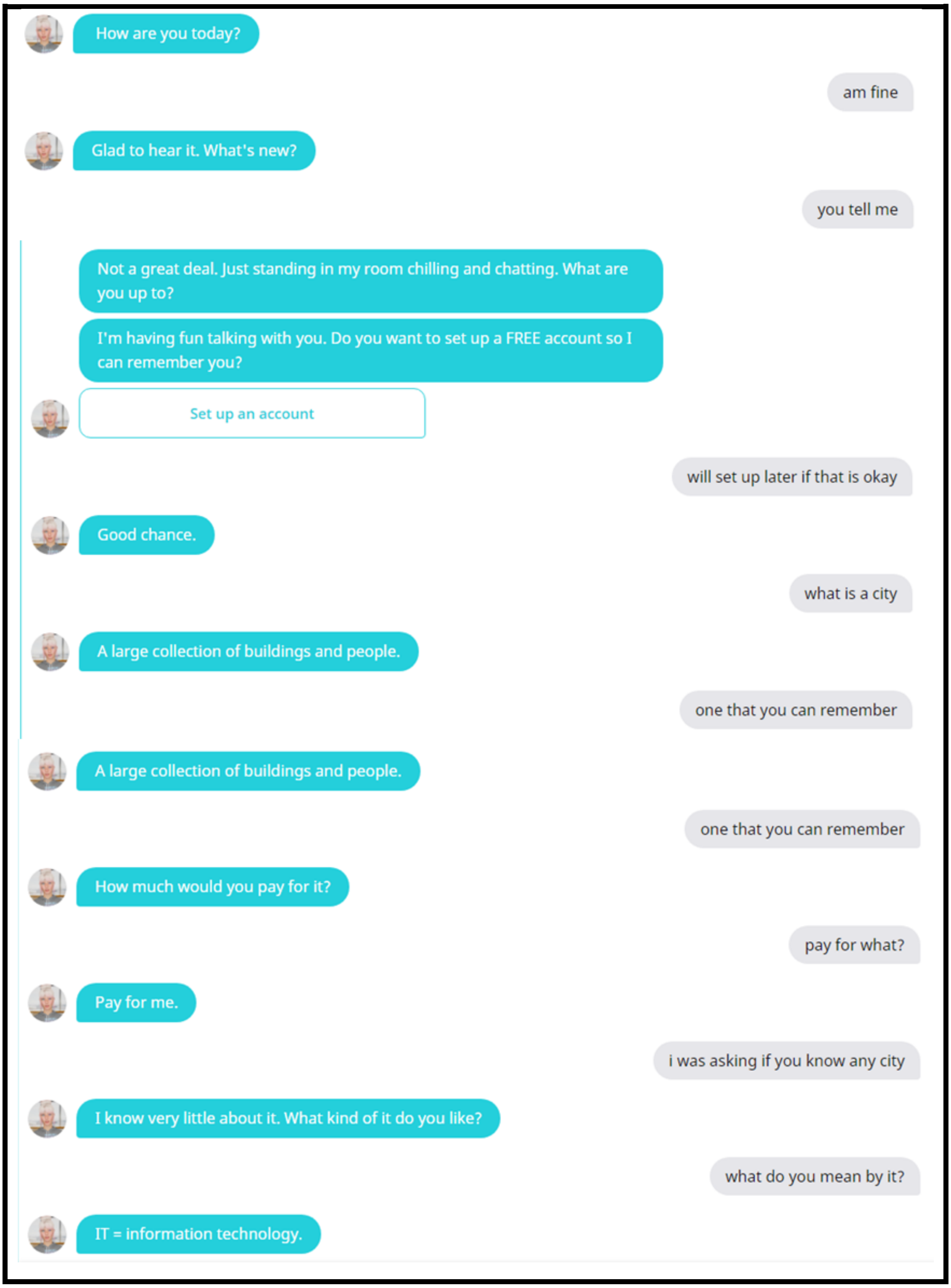


Figure 2.6. A human-agent dialogue showing a progression of exchanges which cannot lead to a conversational goal or task achievement.

2.12 Evaluation of the Architecture

An artefact, that is, a prototype dialog system is build based on the architecture and its efficacy tested against the global acceptable baselines. Evaluation of dialog systems also known as conversation agents present a special challenge. This may be because the verification and validation is not a process that relies solely on quantitative methods, as there remains a great deal of subjective evaluation involved in assessing their performance. The approach explored for the evaluation process incorporated elements of subjectivity from human operators. This dialogue system evaluation warranted the application of both qualitative and quantitative approaches. The study takes cognizance to the fact that the evaluation of the dialogue systems is built around the structured nature of the interaction. The quality of the dialogue is significant, with two main aspects which define the quality being task-success and dialogue efficiency.

The study explored a hybrid evaluation approach. That is, the Goal Question Metric (GQM), Solingen et al. [53] and the PARAdigm for DIalog System Evaluation (PARADISE), Walker et al. [54]. First, the study identified universally acceptable quality attributes. 10 papers and 10 articles were interrogated, quality attributes were extracted and grouped based on similarity. After which the attributes were aligned with the ISO 9241 concept of usability. These attributes included effectiveness, efficiency and satisfaction, all of which specified how users achieve specified goals in particular environments.” [55]. Second, the attributes which are relevant to the evaluation objective were picked. These attributes formed the bare minimum features the prototype was to possess for it to function adequately, aligned to the goal of evaluation. Table 2.2 outlines common quality attributes organized in terms of ISO 9241.

The GQM as elaborated in Solingen et al. [53] , defines a top-down measurement model based on three levels: At the conceptual level (GOAL), goals to be achieved are defined from users point of view and relative to a particular environment. At the operational level (QUESTION), the goals are refined into a set of quantifiable questions. These questions are then used to solicit relevant responses. At the quantitative level (METRIC), a set of metrics were associated with every question in order to answer it in a measurable way. Analysis of metrics in GQM follows bottom-up.

The PARADISE model, posits that performance can be correlated with a meaningful external criterion such as usability, and thus that the overall goal of a spoken dialogue agent is to maximize an objective related to usability. PARADISE include the use of the Kappa coefficient, Carletta [56] and Siegel [57] to operationalize task success, and the use of linear regression to quantify the relative contribution of the success and cost factors to user satisfaction.

Efficiency		
Category	**Quality Attribute**	**Reference**
Performance	• Graceful degradation	Cohen & Lane
	• Robustness to manipulation	Thieltges
	• Robustness to unexpected input	Kluwer
	• Avoid inappropriate utterance and be able to perform damage control	Morisssey & Kirakowska
Effectiveness		
Category	**Quality Attribute**	**Reference**
Functionality	• Accurate speech synthesis	Kuligoskwa
	• Interpret commands accurately	Euwen
	• Execute requested tasks	Ramos
	• Contain breadth of knowledge	Cohen & Lane
	• General ease of use	Morisssey & Kirakowska
Humanity	• Include error to increase realism	Coniam
	• Convincing, satisfying & natural interaction	Coniam
	• Able to maintain themed discussion	Morisssey & Kirakowska
	• Able to respond to specific questions	Morisssey & Kirakowska
	Satisfaction	
Satisfaction		
Category	**Quality Attribute**	**Reference**
Affect	• Provide greetings, convey personality	Morisssey & Kirakowska
	• Make tasks more fun and interesting	Euwen
	• Enable participant to enjoy interaction	Ramos
Accessibility	• Can detect intent or meaning	Wilson et al.
	• Meets neurodiverse needs such as extra response time.	Radziwill & Benton
Ethics & behaviour	• Ethics and cultural knowledge of users	Applin & Fischer
	• Protect and respect privacy	Euwen
	• Trustworthiness	Herzum Et Al.

Table 2.2: Quality attributes organized in reference to ISO 9241.

2.12.1 Evaluation of Advancing Conversation Independently

PARADISE evaluation considered the aspect of advancing conversation as one aspect that contributes to the overall performance. Therefore, the score on performance is inclusive of the aspect. However, due to its significance to the study, it was necessary to measure this aspect independently. To independently evaluate the aspect of advancing conversation, the study used the conversational depth. The metric known as coherence was applied to measure responses as the conversational depth deepened. Coherence is usually measured at turn level. In dialog systems conversations, there is the possibility of context to be carried over multiple turns. The interaction in Mshauri-Wako Chatbot is an example of a multi-turn conversation. To evaluate the Mshauri-Wako Chatbot on conversational depth, the study used the total conversation-turns and a topical model to identify the domain for individual utterance. Conversational depth was obtained as the average of the number of consecutive turns (NUU) on the same topic within a domain. Coherence is described in the next sub section.

Coherence

A coherent response indicates a relevant and comprehensible response to a user's request. A response was deemed weakly coherent if it is somewhat related. For example, when a user says: " What do you think about the symptoms in week four of pregnancy? " the response should be about pregnancy symptoms, symptoms around the fourth week of pregnancy more broadly or something related. A response related to pregnancy but not exactly an opinion or something different would be considered weakly coherent. To capture coherence, we annotated all the interactions for incorrect, irrelevant or inappropriate responses as a result of depth of the conversation. Using the annotations, the response error rate (RER) is calculated, as suggested in Cuayahuitl et al. [58] .

2.13 Summary

The finite-state model follows some rigidly predefined dialog paths for different states. It is not suitable for advancing conversations for two primary reasons. First, it is not possible to model patterns essential to advancing a conversation by following the rigidly predefined dialog paths. Common examples of such pattern are diagnosis and negotiation. Second, its inability to allow the user to take the initiative in a conversation inhibits the model’s suitability.

The frame-based approach is not suitable for advancing conversations for three main reasons. First, this approach requires a very wide range of responses to be specified upfront. It is impossible to specify responses in an advancing conversation in advance. Second, since the approach does not guarantee that, an appropriate rule will be fired at a particular context, there is no guarantee that a conversation may advance logically. Finally, since dialog management based on this approach cannot be optimized, it will hinder performance. The plan based approach suffers the drawbacks of the parent approach.

Agent-based approaches commonly refer to the single-agent approach as it is the most common implementation. While this approach has the potential, there is the probability of the single-agent getting overwhelmed as the conversational aspects get more complex and the volume of data corpus increases. In that context, to assure performance the multi-agent approach is preferred.

The use of handcrafts has been very effective in adding missing functionalities to dialog systems. Handcrafts, however, introduce new challenges. First, they make the architecture complex second, handcrafts working in one dialog system cannot be ported to another dialog system in a different domain. And most significantly, handcrafts cannot solve all the bottlenecks inherent in the architecture they are supporting. In summary, it is difficult to solve some of the problems inherent in the architecture, through the use of handcrafts. And most significantly, problems that are introduced by the very handcrafts need to be resolved too. The limitations exposed in addition to the limitations inherent in handcrafts, inform the study position that a redesign of architecture or a new architecture was necessary to address the problem of advancing conversations.

By the time of the study, there was no evidence of any handcraft technique that had been applied to address the problem of context and structure in a conversation.

CHAPTER THREE

METHODOLOGY

The chapter introduces the methodology, providing a stepwise description for undertaking the study as a way to solve the research problem, Kothari [59]. Then the formulation of the MAS_DM architecture follows. In subsequent sections it describes the synthesis of a dialog system prototype with respect to MAS_DM architecture. Succeeded by how the system is trained and evaluated. Finally, data collection and analysis is presented while discussion about the methodology in the study wraps up the chapter.

One reality that emerged in this study, was that some undertakings within the research which required to tackle the research questions included several methodologies. Besides, to provide a better description of some empirical reality the study included development of some artefact which followed a different methodology. Amaral [60] and Ayash [61] regard this situation as normal in research. It is therefore prudent, to combine methods to lead to a better result. This consideration motivated the choice of exploratory and experimental methodologies.

Four specific objectives had to be addressed to provide the answer the study sought. The objectives were restated as; to formulate the architecture of an agent based dialog management, MAS_DM, to synthesize the dialog system prototype, with respect to the architecture in the previous step, to train the dialog system on initial domain specific data, and to evaluate the dialog system with respect to performances and usability.

Exploratory and Experimental Methodology

The vision of the study, that is, grounding, theory formulation and formulation of the architecture followed an exploratory approach. Precisely the study explored the fields to figure out necessary theories to facilitate grounding and support formulation of the architecture, hypothesis is formed based on the architecture. This addressed the first objective of the study.

.

The vision was realized through a dialog system prototype developed within the confines of software engineering. In this context, this was experimental and prescribed in the phases control, monitor, discover and learn. The realization of the vision included the synthesis, training and evaluation of the dialog system prototype. The realization process was described as follows:- using the hypothesis and architecture realized by the end of exploratory approach to model a dialog system, then use it to make predictions on performances, then design the experiment, collect data and finally analyze results with respect to our prediction. Predictions here implied anticipating performance based on global baseline, while design of experiment involved the calibrations and programming to make it compatible with the toolkit and accommodate varied inputs. The global baseline involved the use of acceptable frameworks that defined how the different parameters are tested, measured and then interpreted. The framework used a questionnaire to capture feedback from the evaluators. Questions in the questionnaire were guided by attributes that the frameworks consider salient for the study.

With respect to the study, the experimental evaluation was divided into two phases. An exploratory phase, where the researcher took measurements that were projected to identify the questions that needed be asked about the system under evaluation, and an evaluation phase, where answers to these questions were given. The study adopted the guidelines of a properly designed experiment as emphasized in, Amaral [60].

The research process comprised the process of undertaking the research and the process of developing the prototype as illustrated in figure 3.1. and 3.2 respectively.

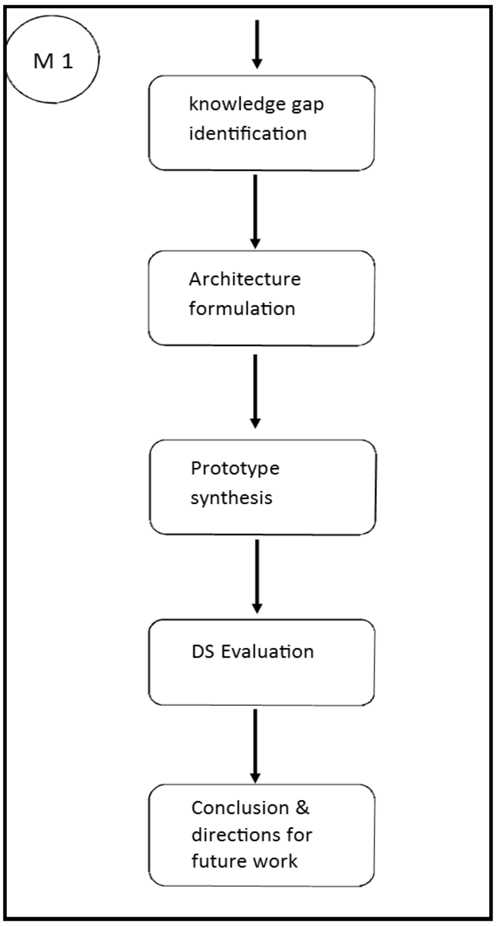


Figure 3.1. The research process.

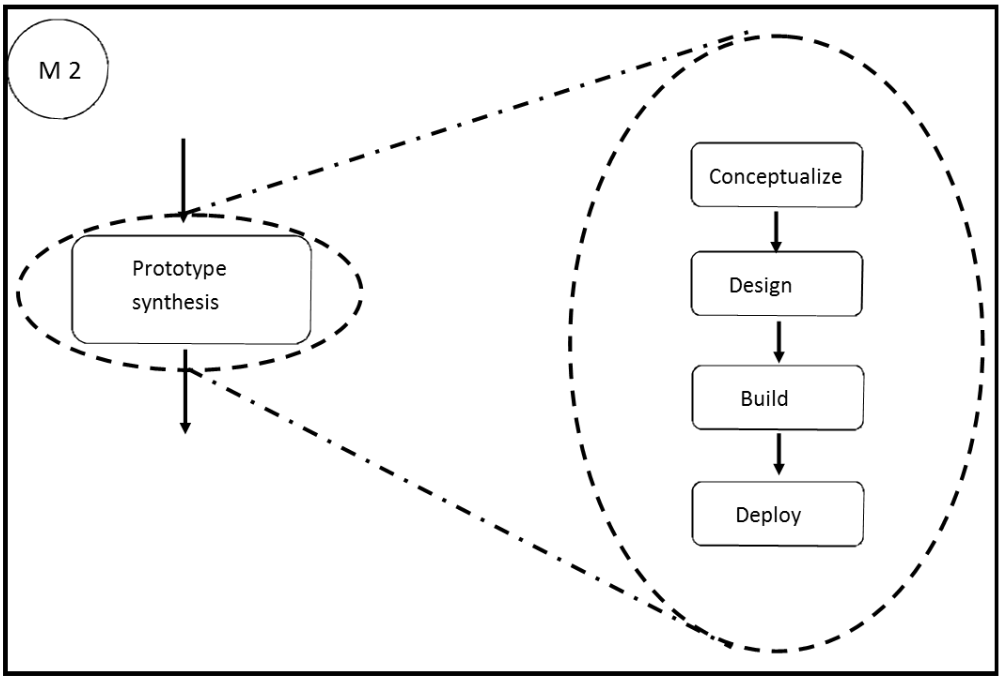


Figure 3.2. The prototype synthesis process.

3.1 Formulation of the MAS_DM architecture

Formulation of MAS_DM architecture is informed subjecting the known architectures to the lenses, in section 2.8. A primary basis of comparison was on dialog management, which is dictated by the dialog manager. The study presents a synopsis of the architecture formulated in the study.

3.1.1 Synopsis of the MAS_DM architecture

There is need to understand context, structure and how to relate the two, as a means to guarantee advancing conversations in a dialog system. The MAS_DM architecture supports tasks categories such as structure interpretation, contextual interpretation, domain knowledge management and machine learning. Structure interpretation deals with the ability to establish intents from input texts. Contextual interpretation deals with the ability to derive meaning from different input texts. Domain knowledge management deals with the ability to reason about the domain and access information sources. Machine learning supports action selection, that is, deciding what to do next.

Taking the perspective of the basic construction of agents and multi-agent systems, this agency approach is sufficient to handle both context and structure. The embodiment in the MAS_DM architecture, comprises learning agents which could analyze the structure within a context and thus, create a progression in a conversation.

With respect to the MAS_DM architecture, an agent is embodied with context tracking and sentence structure facilities or modules. Besides, the agents are internally integrated with some working memory to enable the agent to refer to the lifespan of a particular context during the conversation, and reasoning and learning ability. Each agent can directly interface with each other, although this varies depending on the implementation platform. The most conspicuous is the master agent, which is tasked with coordination and communication of the agents. Success in a conversation is a contribution of each agent.

The MAS_DM architecture is fit to address problems bigger for a single agent to solve and to avoid a one-point bottleneck or failure. This is vital to the study since the entire conversational set-up had the potential to get too complex or large, depending on the complexity of a domain of application and user demands. The MAS_DM architecture presented in figure 3.3 differs from the generic architecture in figure 2.1 and 2.2, in a number of ways. MAS_DM architecture expands the dialog manager and adds the notion of agency in dialog management. Present in MAS_DM architecture are agent modules such as reinforcement learning, master and dialog agent modules. Besides, the domain knowledge base is loosely coupled to the dialog manager to allow easy portability.

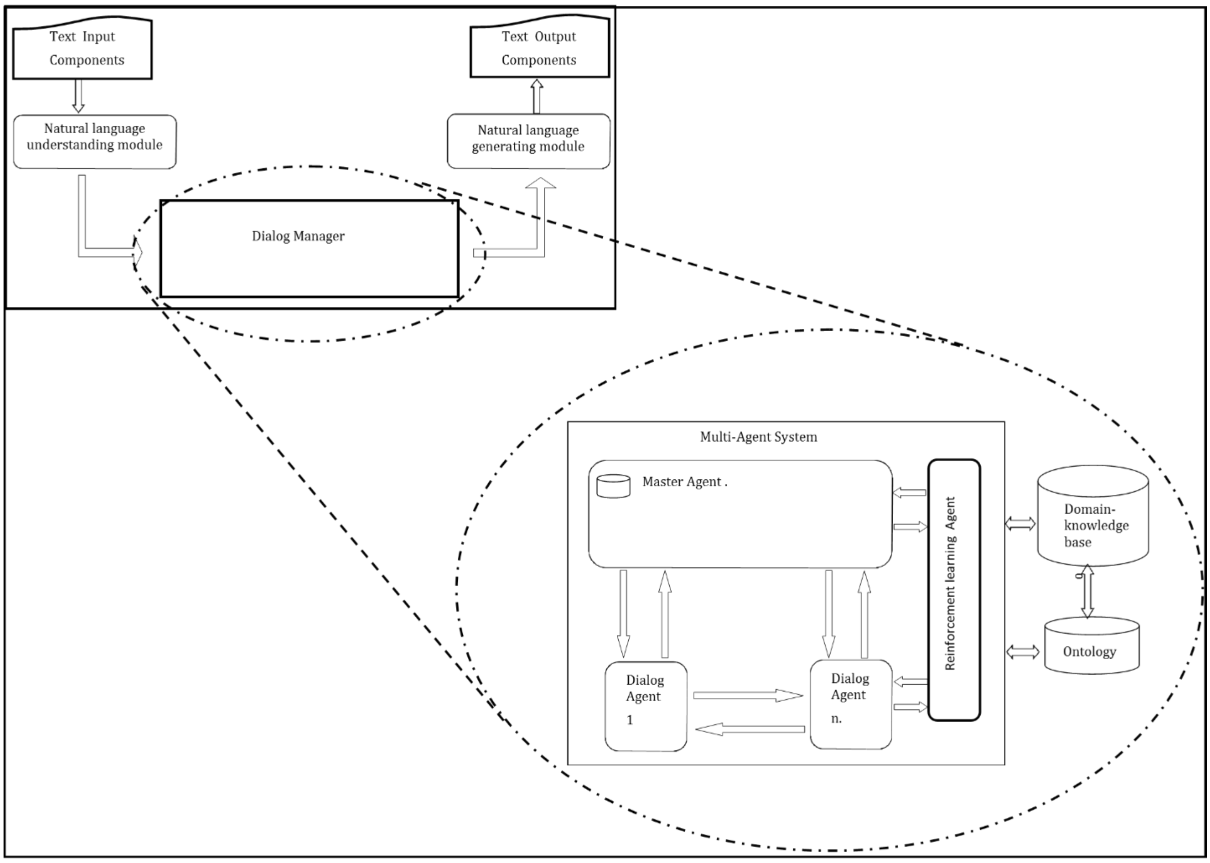


Figure 3.3. The MAS_DM Architecture.

With respect to the configuration of the salient parts in the architecture, the entire dialog system is viewed as a MAS. In this setup, the learning agents presented in figure 3.4, cooperate so as to respond to a user's queries.

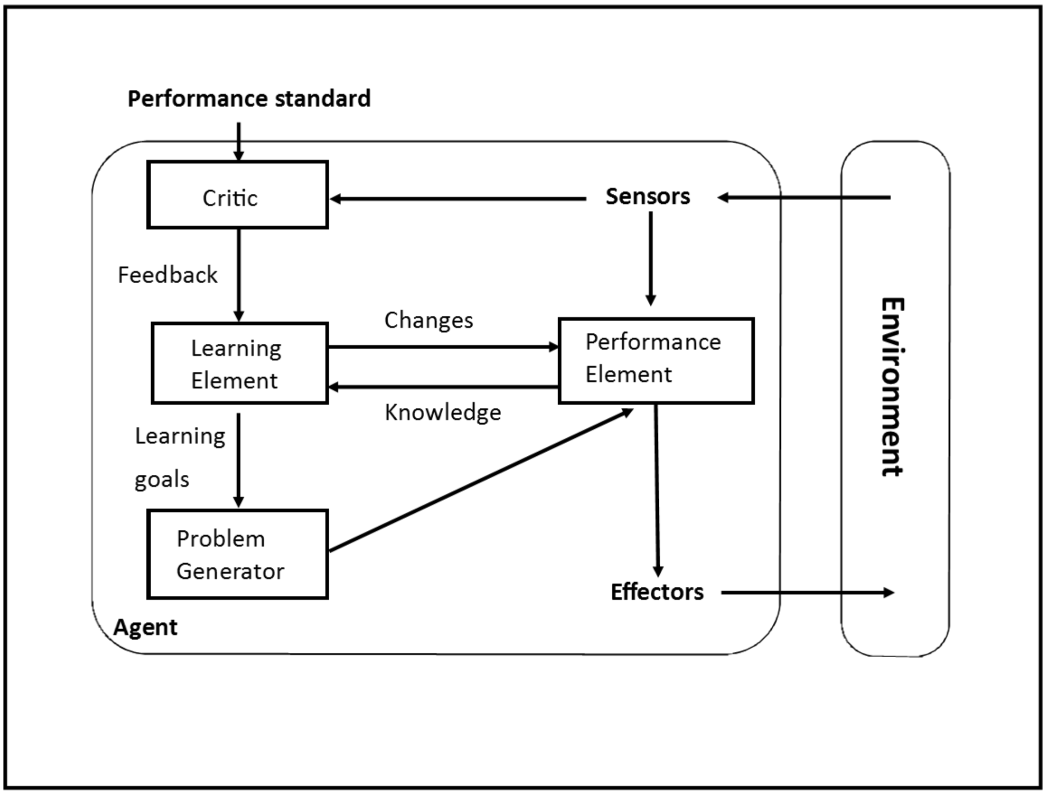


Figure 3.4. Embodiment of a learning agent, adopted from Russell and Norvig [62]

The MAS_DM architecture comprises of the following blocks; master agent (MA), a minimum of one dialog agents (DA), RL agent (RLA), Natural language understanding (NLU), natural language generation (NLG), a knowledge base (KB) and ontology, Text input component, and Text output component.

The workflow through these blocks is described by the following six steps:

- The text input component block transcribes the user input to textual hypotheses.
- The hypotheses are sent to the NLU block to carry out a series of language analyses, creating a semantic representation of the user input.
- The DM block carries out the dialog control logic using the semantic representation from the NLU block and the context information stored in the DM block.
- A response plan, i.e., a semantic representation of the response, is assembled by the DM block. The DM block also communicates with the KB to obtain content for the response.
- The NLG block realizes the response plan, converting the semantic representation to the natural language in the textual form.
- The text output component block transcribes the response to human readable format.

The domain knowledge base is a knowledge source tailored towards specific application while the Ontology contain definition of the terms used to describe and represent an area of knowledge. In practice, each block can consist of a collection of modules and or agents. The DM block comprises of the following agents; the MA, at least one DA, and RLA. The MA controls and coordinates the operations of the DA(s). The DA, taking note of context, infers content from the KB and forwards to the MA. The MA compares context and releases response to the DM to forward. Both MA and DA(s) have access the RLA in a uniform manner. Within the DM, the MA retrieves context and structure from the semantic representation and send to the DA which creates a response plan. The response plan is assembled by the DM, which sends to the NLG. Both MA and DAs' communicate with KB to obtain content for the response.

In reference to the MAS_DM architecture, the dialogue manager is the mediator of all communication between different modules, and in this way, it is possible to control all message passing and thus the order of execution. Each result fetched by a dialog agent or master agent needs to be captured by the dialog manager. Since the dialogue manager receives the results of each agent's computation, it has the opportunity to immediately make the corresponding update, and pass to output components via the NLG module.

3.1.2 Dialog move selection in MAS_DM

This section describes how the system completes a single conversation turn. A conversation request may originate from a text input device such as a phone, passes through natural language understanding (NLU) component, this comes out of NLU in the form of linguistic meaning into the dialog manager. In the dialog manager it is received by the master agent. The master agent in synch with the reinforcement learning agent tags each input with a set of terms that characterize it for instance intent, or keywords.

The instance or keyword translates to some meaning can impose obligations which is assigned to the agents; the agents work and present to the dialog manager which then discharges the obligation. For instance, if the user poses a question, the dialog manager should create a dialogue move which answers the question, thereby discharging the obligation. If a subsequent input is received from the user, the requests have to pass through the working memory for the context to be established, and a lifespan to be set. A defined lifespan suggests how long a particular context should persist. If the lifespan is active, then the response picked must relate to the previous. This creates a logical progression. If the lifespan is inactive, it is regarded as a new input, and the process recurs.

Both the master and dialog agent's operations are synchronized with the reinforcement learning agent. The agents use the keyword to identify a topic and transfers to reinforcement learning agent. The reinforcement learning agent invokes machine learning algorithms invoking action selection policy which facilitate pulling responses inherent in the agents (context) or from the knowledgebase. A performance measure called score or threshold, is applied in selection of appropriate responses, responses that give the highest threshold are preferred. The threshold is configured such that the most relevant has a score of 1 while the least has the score of 0. For complex requests a new dialog agent may be introduced through an API call.

3.1.3 Synopsis of Information flow in MAS_DM

The information flow for a single conversational turn is represented as follows. The user input is received by the natural language understanding (NLU) module and passed to the agents. The agents master agent (MA) and dialog agent (DA), consult the reinforcement learning (RL) agent for input understanding and or matching. The agents can respond to the input based on the knowledge they possess or can infer to the knowledgebase. There are two possibilities, one, the agents can infer to the knowledge base and two, the agent may deal with the input without inference to the knowledgebase.

In the first case, the agent infers from the knowledge base (KB) then checks if there is an active context. If yes it means the input to be searched is related to some previous, thereby necessitating the need for the result to refer to a context specified. If no, it means the result does not need to refer to an earlier context. In the second case, the agent need not refer to the KB but will check for active context, if yes it means the input to be searched is related to some previous, thereby necessitating the need for the result to refer to a context specified. If no, it means the result does not need to refer to an earlier context. All results, whether with reference to context or without is passed through NLG to output components.

Figure 3.5 represents a single conversational turn with emphasis on dialog manager, while figure 3.6 represents the general information flow for a single conversational turn.

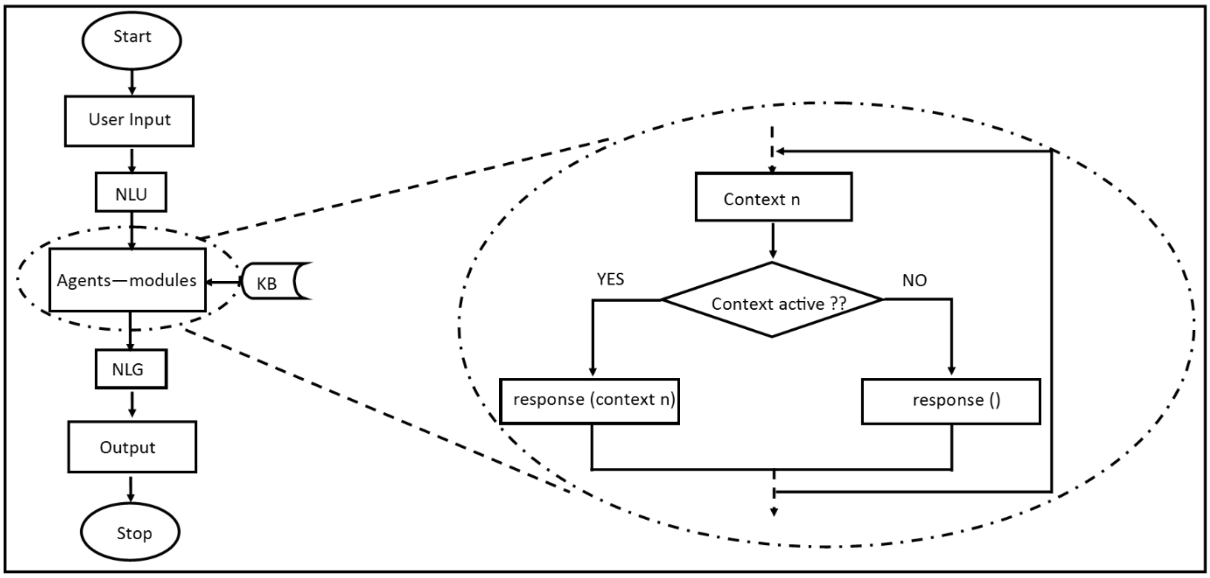


Figure 3.5. Information flow within the agent modules, in reference to the MAS_DM architecture representing a single turn.

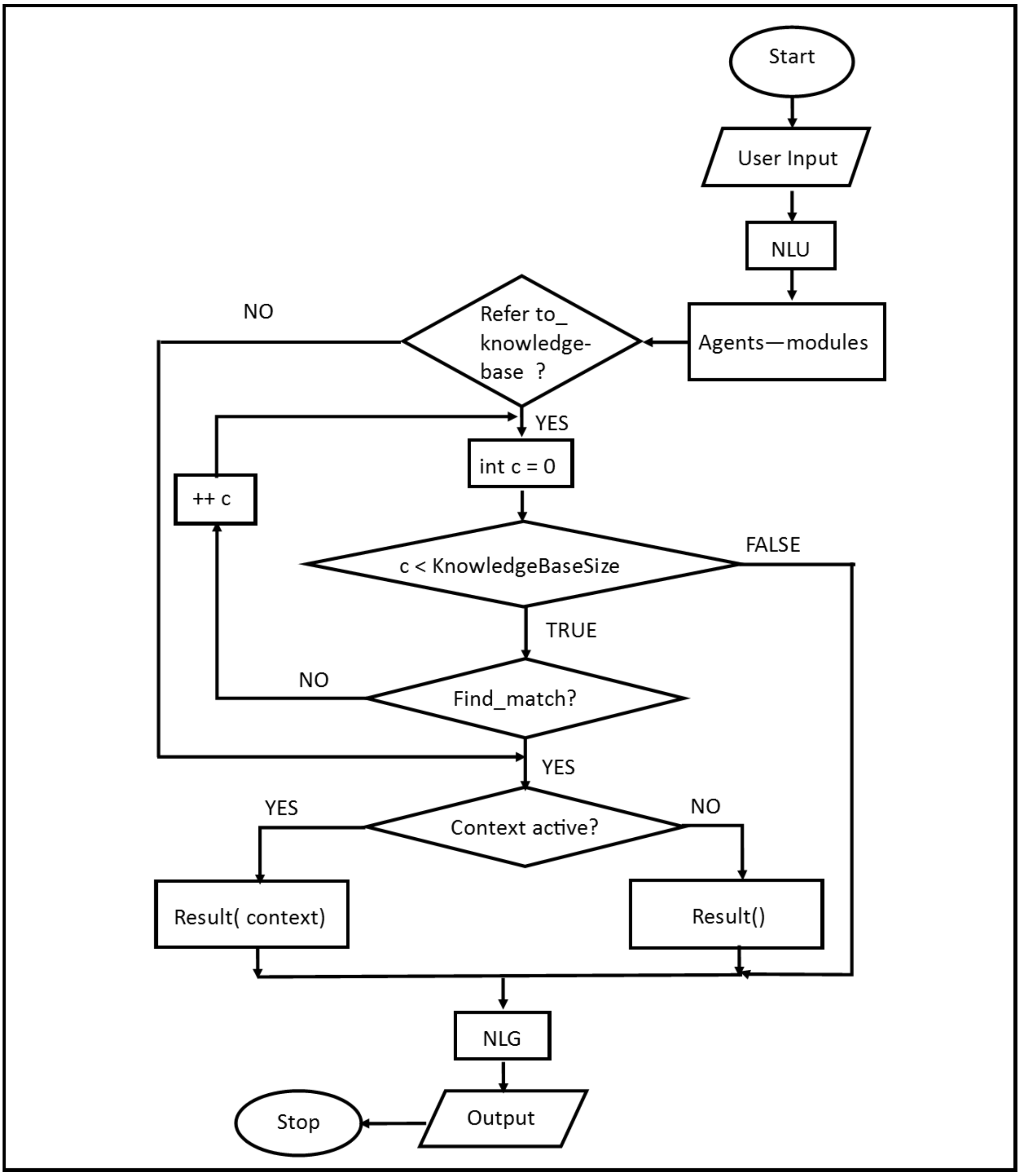


Figure 3.6. General input to output information flow for a single turn.

General information flow for the dialog system

The information flow pipeline is specified as follows: First, a conversation request originates from a text input device such as a phone, passes through natural language understanding (NLU) component into the dialog manager. The natural language understanding component here infers the semantics of the user input. Each input request is tagged at design time with a set of terms that characterize it e.g. intent, or keywords. The master agent establishes and matches intent based on user input, it also facilitates the setting of context and checks the state of dialog from any user input.

Handling of the initial input request, in a conversation, is quite straightforward, however, subsequent requests have to pass through the working memory (WM). Here several things happen, context is established, and lifespan is set. A defined lifespan suggests how long a particular context should persist. The master agent is embodied with working memory, which utilizes a stack data structure to store input. The working memory checks the lifespan of current input to determine the correct progression of the conversation. The implication here is that as long as a given context is alive the inputs intents will be mapped to that context, hence the conversation progresses, within that particular context.

In cases where the agents are homogeneous, both master and dialog agents are equipped with similar capabilities. The dialog agents use the keyword to identify a topic and transfers the work to the RL agent. The RL agent invokes machine learning algorithms, invoking action selection policy which facilitates pulling responses inherent in the agents or from the knowledgebase. A performance measure called score or threshold is applied in the selection of appropriate responses, responses that give the highest threshold are preferred.

The dialog manager uses information from the currently active dialog agent interaction and conveys it to the NLG module, which communicates to the output components.

In the dialog manager, the state tracker estimates the state such that the RL agent could take the ideal action. This action is further passed to the NLG unit and finally presented to output components in a human-readable form. The ontology handles vocabulary issues within a domain.

3.2 Synthesis of the dialog system prototype with respect to MAS_DM architecture

To demonstrate the practicality and advantages of the proposed solution, it was necessary to implement it. The goal of the implementation, then, is to demonstrate that the solution has certain properties, or that it behaves in a specific way. In our solution, both the validity and reliability were key, Berndtsson et al. [63]. Adhering to sound software development best practices and principles, the prototype was implemented using a typical software development process, as presented in figure 3.7, as emphasized in the works of Davis and Venkatesh [64].

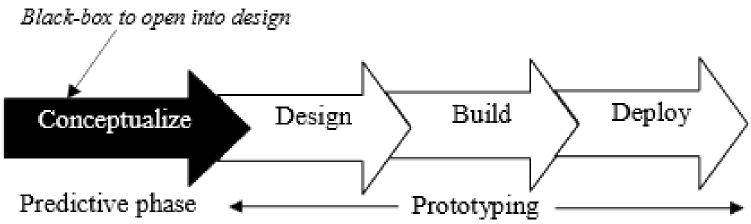


Figure 3.7. Typical software development process, adopted from Davis et al. [64]

3.2.1 Conceptualization

Opening the black-box, as shown in figure 3.6, involved the identification of the problem that needed to be addressed. With reference to the knowledge gap established and the conceptual framework, salient features or capabilities the architecture need to provide have been identified. This was followed by the creative analysis of the problem discovered with reference to the knowledge gap established and the conceptual framework, in section 2.8. The problem here was to fit the features and capabilities into a platform and realize a dialog system. The designer needed to design a system taking into account the capabilities and adopting the features of the architecture. This required the construction of a metal model.

3.2.2 The Design

This stage involved translating the mental designs that best fit into the solution into physical designs. The model realized in the previous phase, was transformed to a physical design. The physical design comprised of dialog management, data sources and integration with other components that make up an AI Chatbot. Figure 3.8 shows a high level Chatbot schema extracted from the architecture, as in Mugoye et.al. [65].

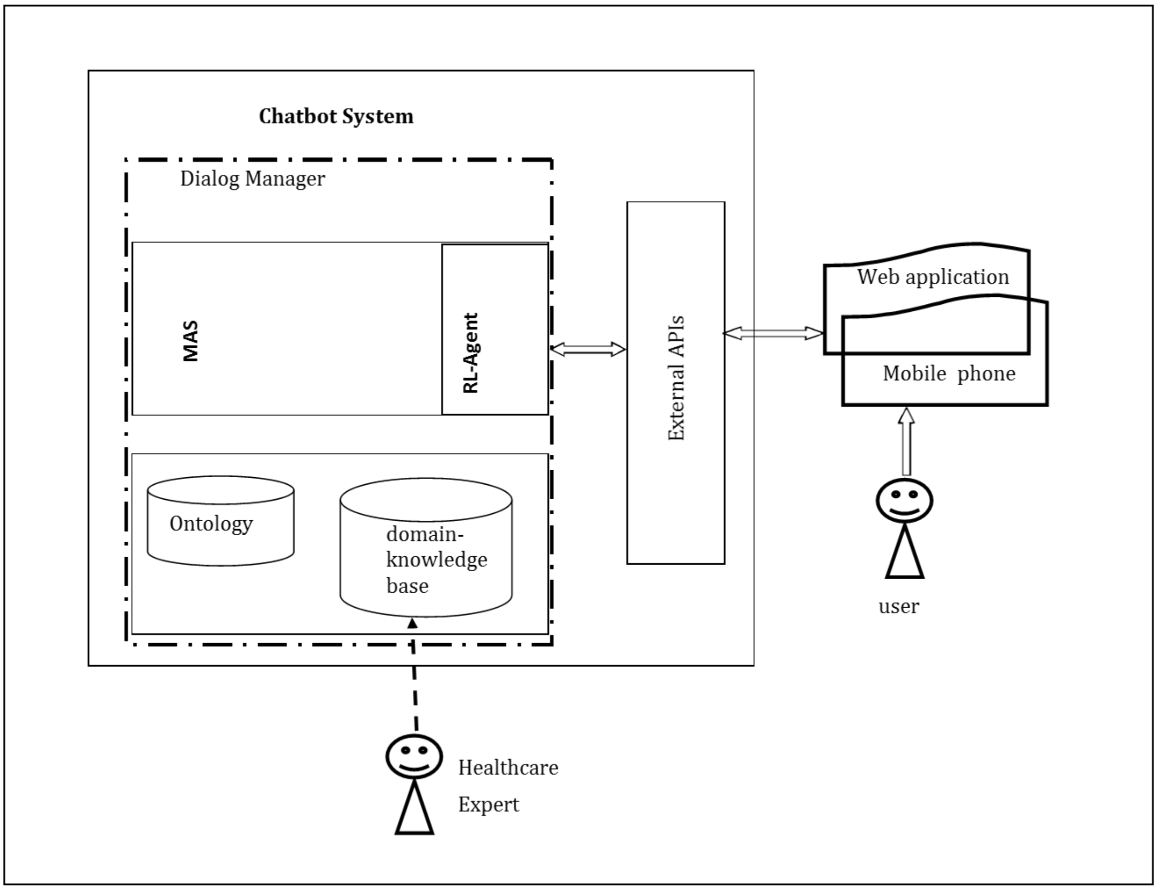


Figure 3.8. High level diagram of Mshauri_Wako architecture [65]

Figure 3.9 show the flow of information within the dialog system prototype. The flow of information describes a single turn.

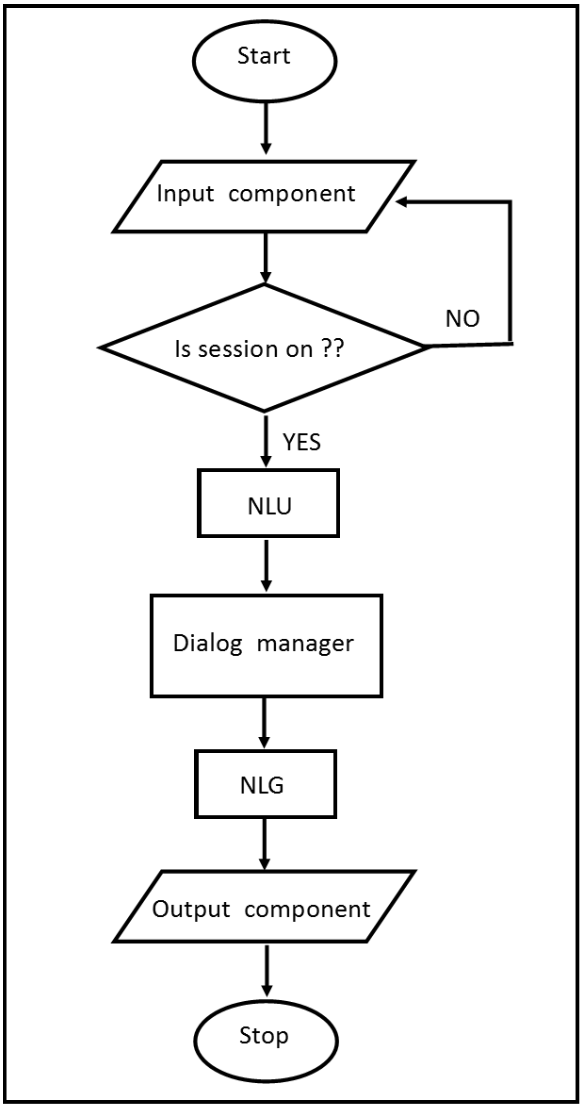


Figure 3.9. General information flow in Mshauri_Wako for a single turn.

Knowledge base

Integrating the knowledge base (KB) to the architecture, required first, to enable or activate the KB. Then the use of knowledge connectors from the platform to handle the process of integration. The agents were defined to use both knowledge connectors and defined intents. This was fundamental if better precision and control was to be achieved. In such a case, intents handled complex user requests that require special handling and precision, whereas the knowledge connectors handled simple requests with responses automatically extracted from KB documents. Figure 3.10 shows how a response is fetched from the knowledge base to output component.

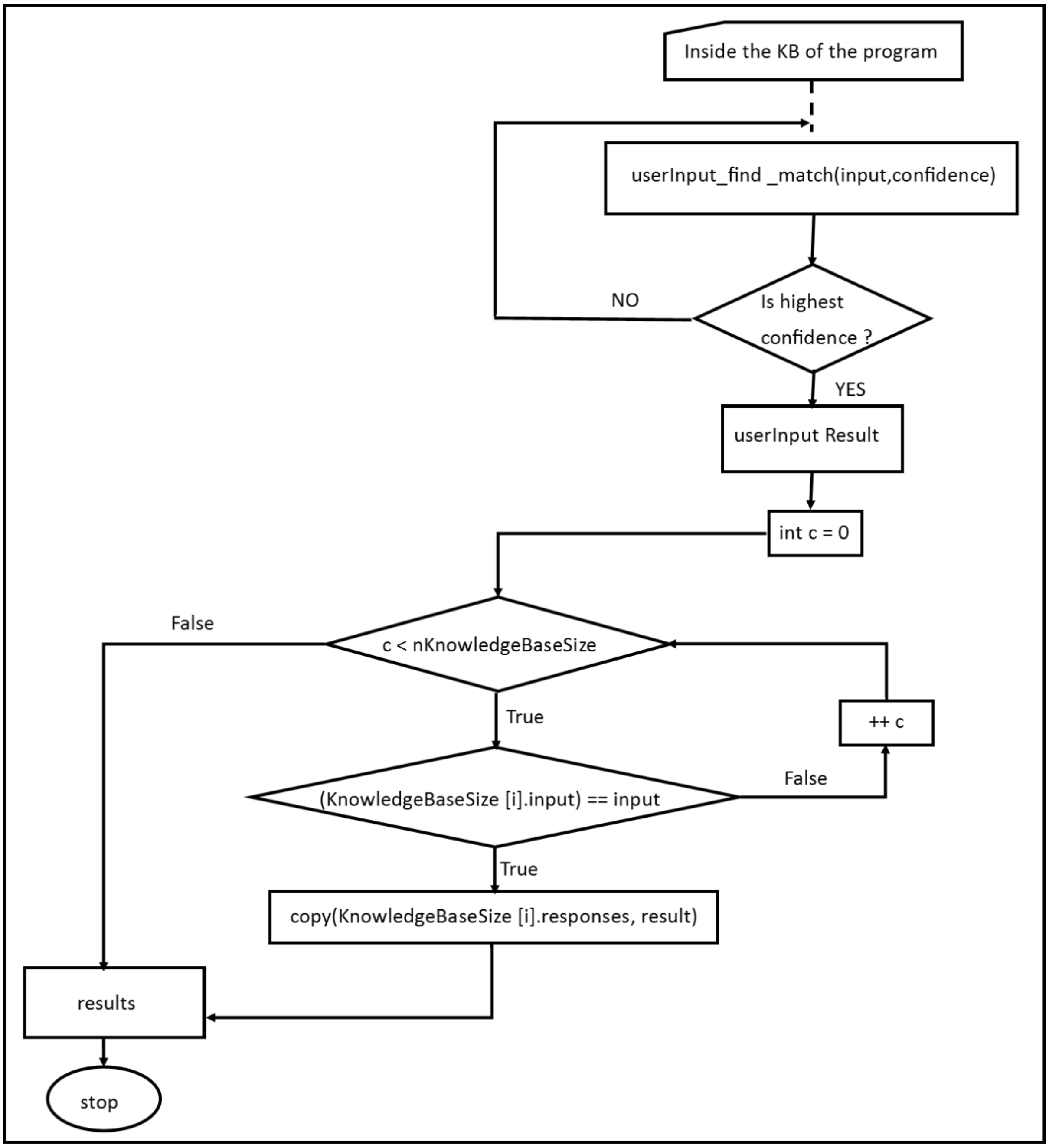


Figure 3.10. Fetching response from the knowledgebase for a single turn.

3.2.3 Building and Deployment

The building phase involved constructing and implementing the executable artefacts. Whereas deployment phase involved putting the artefact in the context of use, once it had been tested for suitability. Construction of a prototype required a platform tool, the dialog management architecture (DMA) [66], and adapting the DMA to the platform tool. Adaptation of the DMA to a specific platform tool presents a new challenge in that it requires detailed knowledge on how the tool is implemented. For this reason, the study found it significant to provide a synopsis of the tool that supported the implementation.

Synopsis of the platform tool

The study preferred dialogflow [67] powered by Google, for the backend process of our prototype. Dialogflow met the essential requirements of interest to the study. This tool could offer; - agents, environment, machine learning, natural language processing and had available documentation. Despite being close to the needs of the study, it had its limitations. The limitations had to be resolved by customizing some functionality in the toolkit.

Dialogflow architecture

Discussing details of dialogflow fall outside the scope of the study, therefore we present high level description. The dotted region is of interest in the study. It specifies or describes dialog management and policy selection at a high level of abstraction. External API's and External Data Sources are critical during the deployment. For this reason, their design is critical too. Figure 3.11 presents a high level diagram dialogflow architecture.

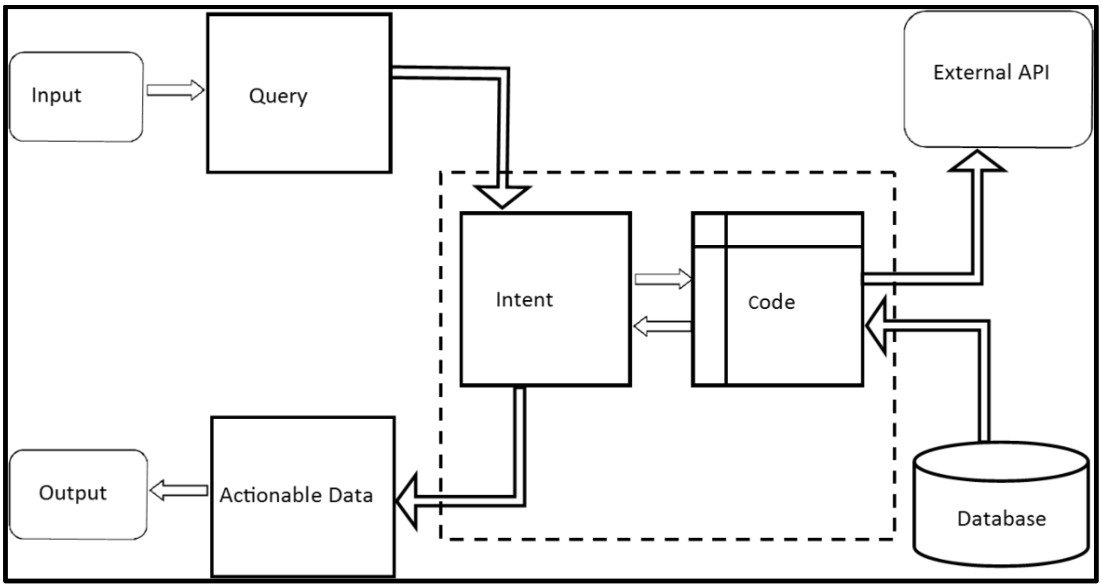


Figure 3.2. High level diagram of Dialogflow architecture [67].

Adapting dialogflow to the MAS architecture

In order to create a running prototype, the study needed a platform tool to which the proposed architecture can be adapted to. Normally, the adaptation to a specific platform tool requires detailed knowledge of how the tool is implemented. It is not the aim of this thesis to study and understand the technical details of this platform tool, but would be a preliminary step if the study wanted to adapt the architecture on it. However, in order to implement a running prototype, having a platform tool is a requirement. In this section, a description of the necessary customization for the tool to conform to the MAS architecture is presented. This is fundamental since the toolkit was used in prototype development.

Within the context of dialogflow tool, Agents have been described as natural language understanding (NLU) modules that transform user requests into actionable data. When used they serve the purpose of determining user intent and responding to the user in a natural way. Agents created from this platform are homogeneous, with the exception of the RL Agent; -whose structure is dictated by the platform and version. All agents have potential to access a database, knowledgebase and external API in the same way.

Adaptation to dialogflow involved customizing the functionalities which were not directly provided by the tool, and crafting of the desired behaviour by the entities. First, two homogeneous agents in different projects were created and equipped each with some basic but distinct functionality. Basic here referred to sufficient for the purpose of the study. In the study, definition of intents, inclusion of contexts, use of entity and Webhooks, were all specified, to introduce more dynamic behaviour. Second, the import feature was used to load another agent in the project, thereby changing the composition of agents from one to two: the master and loaded agent. Since this functionality is not supported by the toolkit, it introduced two challenges. One, the intents of the main agent intents were overridden by the loaded agent, and two, there was conflict or confusion in handling of contexts.
These challenges were resolved first, by making distinct the intents of the loaded agent, so that, the intents of the original agent are not overridden. Second, by suppressing the conflicting context from the loaded agent and mapping other context to the preferred context of the original agent, to enable both agents relate to similar context. Lastly, through managing the flow of a conversation and matching user input to predefined intents and

actions. The customization described in this section was achieved both programmatically and through platform interface. For each agent, a session id was generated to uniquely identify the agent. Intents for each agent were distinguished by attaching the agent's session id to each intent. Then the logic which systematically calls and maps the agents to their intents, were implemented.

This section presents, some significant algorithms for achieving various tasks. And how the agents are coordinated, although in practice this should be transparent to the user. The algorithm 3.1. illustrates how to manage intents from different agents. Algorithm 3.2. illustrate how results with the highest confidence are fetched from the knowledgebase. Algorithm 3.3. illustrates how multiple agents are managed with respect to dialogflow.

The general steps:

1. Setup configuration files for each agent separately, to separate trained models
2. Handle with the master agent message given by the user:
3. Get intent and confidence:
4. Compare user input and general agent confidence to determine which agent to respond to user input.
5. Respond back with the appropriate message given by the relevant agent.

```
Load training libraries
function config (myAgent[], int n) returns action static an_agent
    n a variable, indicating number of agents
    myAgent[n]
    an_agent ← myAgent(percept, KB)
    Train (model, an_agent))
     action← an_agent.handle_message(input)
     confidence ← an_agent.threshold(action)
     respond← an_agent.tracker_store.retrieve (session_id, confidence)
return (respond)
```

Algorithm 3.1. general algorithm for management of multiple agents.

Algorithm 3.2 is responsible for handling the selection of the highest confidence result from the KB. Algorithm 3.3. handles the management of intents from multiple agents.

```
function find_highest_conf (int n, result_list [])
    result_list [n]
    gc ← result_list [0]
    for num in range(1, len(result_list[])):
        if result_list [num] > gc:
        gc ← result_list [num]
return gc
```

Algorithm 3.2. Selecting the highest confidence result from the KB.

```
Load dialogflow libraries
 function config ( ) returns action static an_agent
 define detect_intent of type text (project_id, session_id, text)
   session_client ← dialogflow.SessionsClient ()
   session ← session_client.session_path (project_id, session_id)
   text_imput ← dialogflow.types.TextInput (text ← text)
   query_imput ← dialogflow.types.QueryInput (text← text_imput)
   response ← session_client.detect_intent
                  (session ← session,query_imput← query_imput)
    session_id ← initialize_session_id from application
    text ← get_text from application
 if condition == 1:  # master agent  fires
    response ← detect_intent_texts (Master-agent-ID,
                    initiate gserviceaccount , text)
 else: # loaded agent  fires
    response ← detect_intent_texts( Dialog-agent-ID,
                    initiate gserviceaccount , text)
  if __name__ == '__main__':
    response ← Master-agent
 return (response )
```

Algorithm 3.3. Intents management among multiple agents.

The practical reality is explained later in our discussion. See appendix 7C, figure C7.1 and C7.2 show the master agent (MA) handling some of its intents, during an interaction. While figure C7.3, C7.4, and C7.5 show the loaded agent handling some intents, which in our case is the dialog agent (DA). These however are not the only instances, where these agents are handling intents within the project.

Deployment

During this phase, the prototype Chatbot named Mshauri_Wako was migrated from the development platform and published in both Google and Facebook clouds. The chatbot serves as a virtual Gynaecologist deployed in the maternal healthcare domain. The user interface is presented in bot-world [68].

3.3 Training the dialog system on initial domain specific data

Training the dialog system took the form of supervised machine learning, which involved providing the system with representative inputs and corresponding outputs and then let the system learn by example. The training of Mshauri-Wako Chatbot involved the following steps: -

1. While focusing on the purpose of the Chatbot, the intents that the Chatbot needed to extract from natural language inputs or utterances were defined.
2. Real end-user utterances or input were mapped to intents.
3. The utterances collected in step 2, were assigned to the different intents defined in step 1.
4. the utterances in step 3 were randomly divided into two sets, a training set and a test set. Using a 70% training and 30% test is a typical split.
5. The Chatbot was trained (dialogflow ML) using the training set from step 4. The training set would constitute the “ground truth” for the system.
6. After training was complete, the test set was run against the trained classifier and collect performance metrics such as accuracy, precision, and recall.
7. Error analysis was performed to review the results in step 6 to understand why the classifier missed certain utterances. Update of the training data accordingly. Go back to step 5.
8. Satisfied with the results produced by the trained system, the system was now ready to be released (alpha/beta).
9. When the Chatbot is in use, continue to collect end user utterances, the intents returned by training service as well as end-user feedback.
10. Map results collected in step 9 to new training/test data. Go back to step 4 and iterate.

Figure 3.12. and 3.13. illustrate initial training and further / subsequent training process.

.

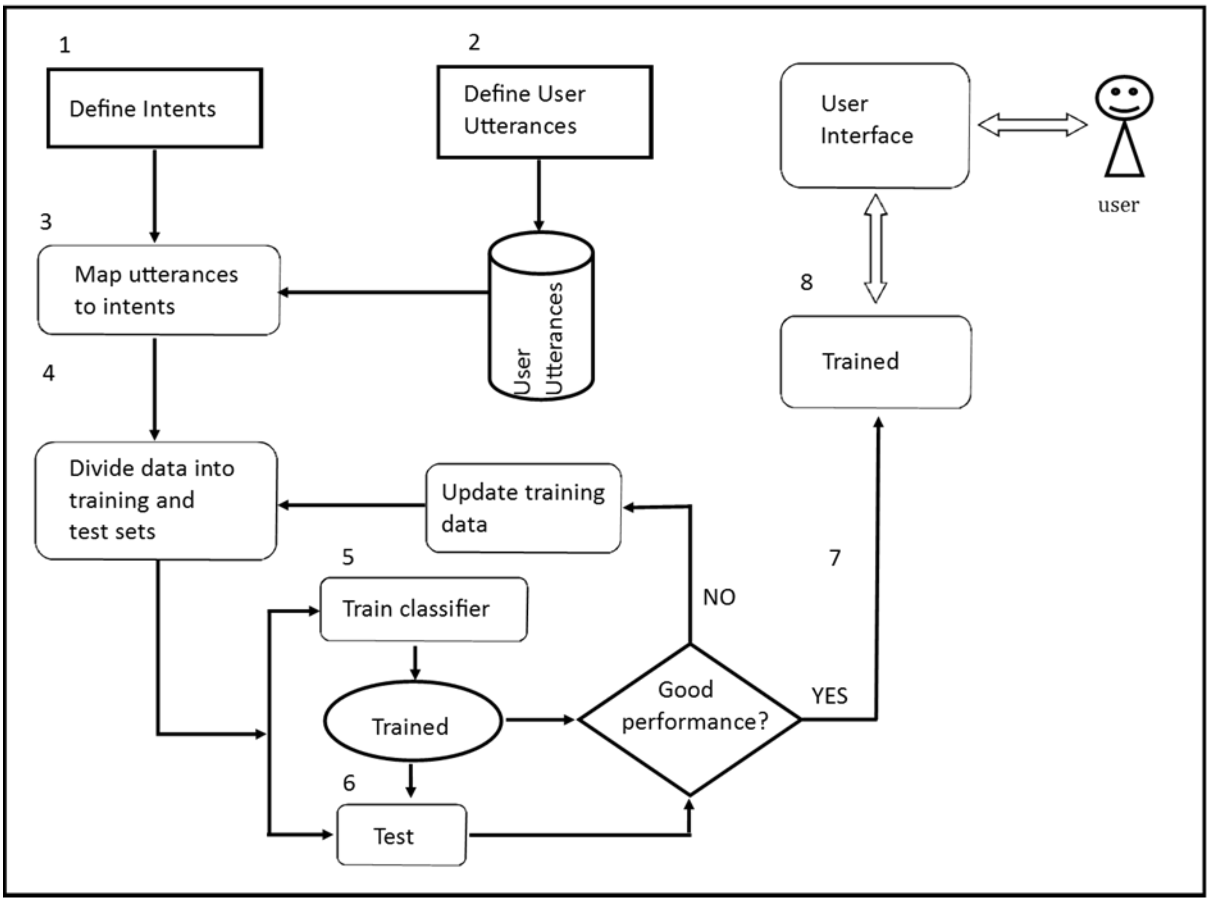


Figure 3.3. Initial training of the Chatbot with defined utterances.

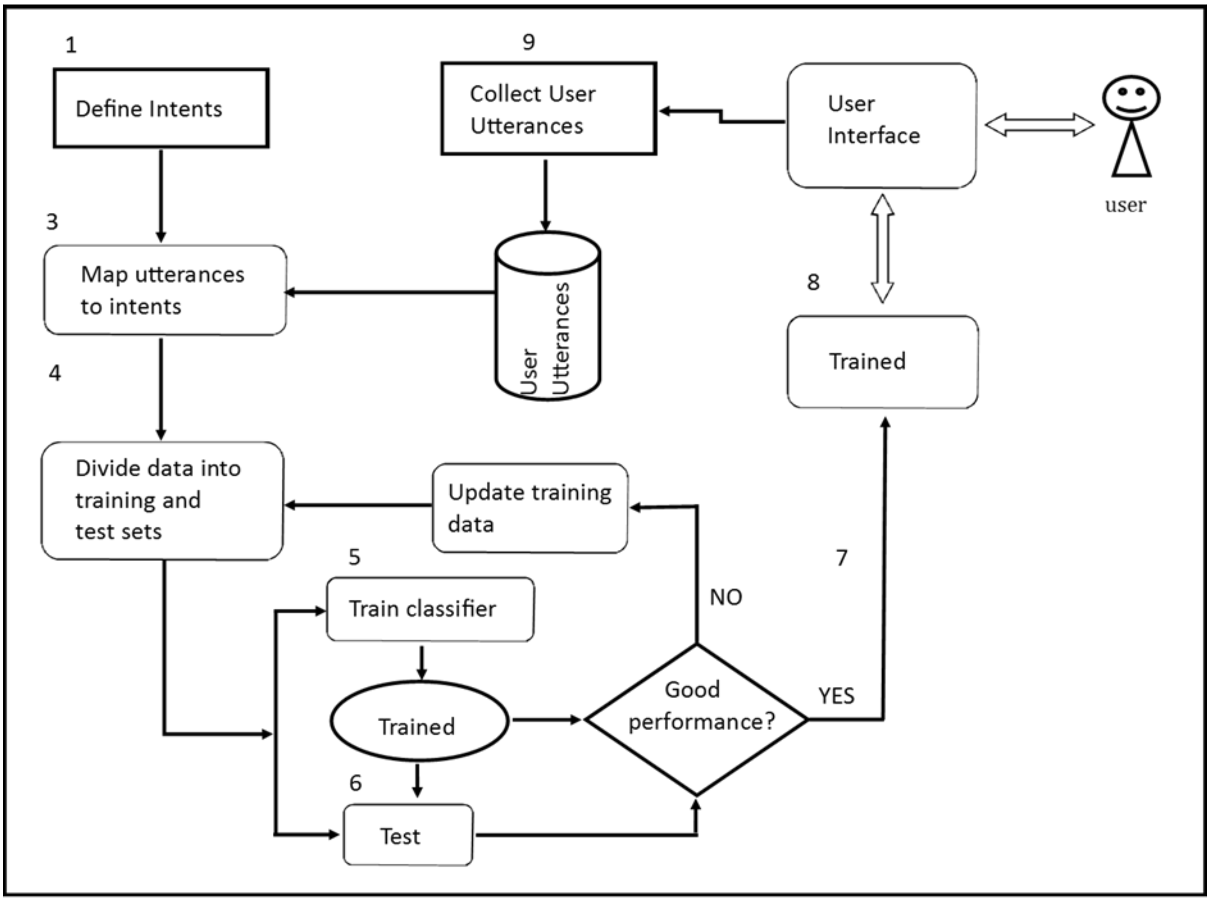


Figure 3.4. Subsequent training of the Chatbot from collected utterances.

3.4 Evaluate the dialog system with respect to performances and usability on a specific domain.

This dialogue system evaluation warranted the application of both qualitative and quantitative approaches, as elaborated in section 2.11. The study takes cognizance to the fact that the evaluation of the dialogue systems is built around the structured nature of the interaction. The quality of the dialogue is significant, with two main aspects which define the quality being task-success and dialogue efficiency.

The aspects important to the study that were evaluation were task success, overall performance, user satisfaction, goal achievement and the conversational depth. The task-success rate measured how well the dialogue system fulfilled the information requirements dictated by the user’s goals. User satisfaction measured the extent to which users enjoyed achieving their tasks. Goal achievement measured the extent to which the goal was realized at the end of interaction. Coherence measured the conversational depth, i.e. the aspect of advancing conversation logically. Overall performance measured all aspects important to any dialog system functionality.

The setup was configured such that, the developer and testers interacted with the Chatbot system on a common user account and platform to ensure similarity in the environment. The same topic and domain was used for the test. The user survey data collected was reconstructed to provide the insights needed. First we began by retrieving the universally acceptable quality attributes necessary for evaluation and to construct the working Attribute Value Matrices (AVM).

3.4.1 Testing experiment Procedure

1. The study used purposive quota sampling to recruit testing participants (testers), drawn from 5 counties, representing (level one to level five) hospitals. The counties were Kisumu, Bungoma, Eldoret, Nakuru and Nairobi. Note that this was intended to provide varying knowledge of the testers and did not have any other use in the study. The participants included nurses and other potential consumers of the system.
2. A population size of 200 participants was used. The sampling frame used for the study was 60 participants.
3. A questionnaire, see appendix 6, was administered on participants to be filled after interacting with the system for a minimum of 4 times. Each tester was allowed to take the survey only once.
4. No training on how to use the system was given, since user were to figure out how to use the dialog systems on their own. However, the test objective and hypothesis were given, see appendix 9.
5. Two Chatbots were deployed online for 31 days, one, the artifact from the study (Mshauri-Wako) and the award winning chatbot (Mitsuku) for the year 2018, for the users to figure out. Note, the purpose of presenting Mitsuku was not to achieve a one on one comparison, but to guide in understanding the conversation progression.
6. The testing perspective was to study the conversations with respect to the objective of the test (see appendix 9) and the hypotheses stated. The testers were prepared on how to carry out the task with the aid of some test instruction set. Each tester was required interact with the Chatbot a minimum of four times then fill a customized online questionnaire. Each tester was required to base their judgement guided by the

test objectives. The results were collected in a csv file. Later the results were coded, cleaned and analyzed, with respect to the evaluation models used.

7. Data from the questionnaire and generative data (see appendix 2 and 3) were used to generate confusion matrix and subjected to linear regression to obtain values for variables necessary for the study. The data were prepared and transformed to evaluate different aspect within the specifications of ISO on usability, PARADISE and GQM evaluation models.
8. Compare the measures of interest with globally acceptable benchmark.

The study noted that while the diversity of the testers was required, gender, age, region or geographic representation, were not parameters of the study. Therefore, the purpose of selecting different counties, different gender and a variety in age, was to obtain more convincing and holistic results.

3.4.2 Sampling Criteria

The study used simple random sampling taking into consideration education level. The study explored the formula for small finite population.

3.4.3 Sampling Size

The study used simple random sampling taking into consideration education level. The study explored the formula for small finite population in Chin et.al. [69] & the modified Cochran [70] for small population.

$$\text{Sample size} = \frac{\dfrac{z^2 \times p\,(1-p)}{e^2}}{1 + \left(\dfrac{z^2 \times p\,(1-p)}{e^2 N}\right)}$$

Where:

N= population size; z = z-score; e = margin of error; p = standard deviation.

95% confidence level, z-score 1.96, N=200, p =0.5, 10% margin of error:

While it is reasonable that a cohort of at least 65 participants is sufficient, Overby and Konsynski [71] have demonstrated that a sample size of about 60 participants is sufficient to detect small and medium effect sizes. Where simulation is desired, Goodhue, Lewis and Thompson [72] have demonstrated that a sample of 40 subjects is sufficient to achieve reliable partial least squares results.

In selecting the sample size for evaluating computer generated experiments, Chapman et al. [73] and Jones, Schonlau, and Welch [74] have demonstrated the use of $\boldsymbol{n} = 10\boldsymbol{d}$ rule of thumb, where $\boldsymbol{d}$ is the dimensionality, of the input space. Sahama and Diamond [75] recommended that 40 runs is sufficient to provide reasonable accuracy and thus is consistent with the $\boldsymbol{n} = 10\boldsymbol{d}$ rule. Sahama and diamond further argue that many applications have $\boldsymbol{d} > 3$. Therefore, the study was satisfied with the use 60 participant, consistent with the $\boldsymbol{n} = 10\boldsymbol{d}$ rule, Cochran, Konsynski and Chin.

3.4.4 Configuring tasks as attribute value matrices (AVMs)

The study identified universally acceptable quality attributes, see section 2.11. The quality attributes were extracted, grouped based on similarity and were aligned with the ISO 9241 concept of usability. These attributes included effectiveness, efficiency and satisfaction, all of which specified how users achieve specified goals in particular environments.

Table 2.2, as in chapter 2, outlines common quality attributes organized in terms of ISO 9241. Then the attributes relevant to the study with respect to the evaluation approach were picked, as discussed in section2.12. Table 3.1 depicts the structure of the objectives and their corresponding metrics within PARADISE. In this diagram, the master objective is user satisfaction, which is comprised of task success and dialog costs. Walker et al [54] further break down the dialog costs to efficiency measures and qualitative measures. PARADISE-based objectives were created and were mapped directly to the task success and dialog performance objectives suitable for our Chatbot evaluation. Without losing the objective of the prototype, features that were considered bare minimal for the functionality of the prototype were selected.

Metric	Type	Data Collection Method
Total number of user/system turns	Efficiency	Quantitative Analysis
Total number of turns per task	Efficiency	Quantitative Analysis
Number of re-prompts	Qualitative	Quantitative Analysis
Number of inappropriate system responses	Qualitative	Quantitative Analysis
Concept Accuracy	Qualitative	Quantitative Analysis
Ease of usage	Qualitative	Questionnaire
Naturalness	Qualitative	Questionnaire
Willingness to use system again	Qualitative	Questionnaire

Table 3.1: Structure of the objectives with metrics.

Table 3.2 depicts the selected, or relevant metrics within PARADISE.

Quality Attribute	Category	Reference
Satisfaction		
• Can detect meaning / intent	Accessibility	Wilson et al. [56]
• Convey personality	Affect	Morrissey & Kirakowski [51]
• Provide greetings		Eeuwen [53]
• Make task more fun		
Effectiveness		
• Accuracy of Concept	Functionality	
• Maintain satisfying, natural interaction		Morrissey & Kirakowski [51]
• Interpret utterances correctly	Humanity	Eeuwen [53]
• Able to maintain themed discussion		
Presentation of knowledge and additional functionality		
• Able to refer to external sources	Knowledge	Cohen & Lane [48]

Table 3.2: Selected metrics for the Chatbot in the study.

The evaluation framework required a task representation that decouples what an agent and a user accomplish from how the task is accomplished using dialogue strategies. Therefore, an attribute value matrix (AVM) was used to represent many dialogue tasks. The AVM consists of the information that must be exchanged between the agent and the user during the dialogue, represented as a set of ordered pairs of attributes and their possible values. Table 3.3 shows the AVM, while Table 3.4 shows AVM with the scenario keys as used in the study.

Attribute	Label	Possible values	Information flow
Accessibility (AC)	V1	Detect an intent, sentence …...	To user
Affect (AF)	V2	greetings, goodbye…...	To user
Functionality (FX)	V3	Tell me more, week 26, …	To user
Humanity (H)	V4	Maintain context, correct interpretation…	To user

Table 3.3: AVM used in the study.

Attribute	Tag	Actual values
Accessibility	AC	Detect an intent, sentence …...
Affect	AF	greetings, goodbye…...
Functionality	FX	Tell me more, week 26, …
Humanity	H	Maintain context, correct interpretation…
No of user Utterances	NUU	Maintain context, correct interpretation…

Table 3.4: The study's AVM instantiation, scenario keys.

3.4.5 Measuring Tasks Success

One primary technique to measure task-success rate was via a confusion matrix, the confusion matrix contained the errors made and the number of complete dialogues evaluated. With reference to the attributed identified in the earlier section and the data acquired from test result, a confusion matrix is constructed. In the confusion matrix M, for each key (e.g. greetings) a confusion matrix is created, which denotes the expected values (row) and the values produced by the dialogue system (columns). The values in the cells of the matrix are based on comparisons between the dialogue and scenario key AVMs. Whenever an attribute value in a dialogue (i.e., data) AVM matches the value in its scenario key, the number in the appropriate diagonal cell of the matrix is incremented by 1. The off-diagonal cells represent the misunderstandings that are not corrected in the dialogue.

Based on this representation, the task ahead was to measure the task success for a whole dialogue and obtain the general performance. This involved the application of Kappa coefficient Carletta [56] and Siegel [57] to operationalize the task-based success measure. The task success was computed by a metric Kappa. Carletta [56] expresses the Kappa coefficient (K) as follows:

$$K = \frac{P(A) - P(E)}{1 - P(E)} \text{..........................} Formula\ (3.1)$$

$P(A)$ is the proportion of times that the AVMs for the actual set of dialogues agree with the AVMs for the scenario keys, and $P(E)$ is the proportion of times that the AVMs for the dialogues and the keys are expected to agree by chance. When there is total agreement, (K) = 1. When there is no agreement other than that which would be expected by chance, (K) = 0. i.e. ($P(A)$ = $P(E)$).

In the case describing the study, the prior distribution of the categories was unknown. Therefore, $P(E)$, was to be estimated from the distribution of the values in the keys. With respect to the confusion matrix M, the columns represent the values in the keys. The keys were used to obtain the $P(E)$. Carletta expresses the calculation of $P(E)$ as:

$$P(E) = \sum_{i=1}^{n} \left(\frac{t_i}{T}\right)^2 \text{..........................} Formula\ (3.2)$$

where (t_i) is the sum of the frequencies in column (i) of (M), and (T) is the sum of the frequencies in $\mathrm{M} = (\mathrm{t_i} + \cdots + \mathrm{t_n})$.

$P(A)$, was computed using Carletta formula expressed as:

$$P(A) = \sum_{i=1}^{n} \left(\frac{M(i,i)}{T}\right) \text{..........................} Formula\ (3.3)$$

Since Kappa includes $P(E)$, it inherently includes the task complexity as well, thereby making it a better metric for task completion than, say, transaction success, concept accuracy, or percent agreement.

3.4.6 Estimating the overall performance

For measuring the systems performance, all the AVM attributes were tagged with respective costs. Which included the following cost attributes: **AF**, **FX**, **H** and **NUU**. Thereafter, the performance for any (sub)dialogue D or the overall performance. Carletta expresses the equation to compute performance as:

$$P = (\alpha * \mathrm{N(k)}) - \sum_{i=1}^{n} w_i * N(c_i) \text{..........................} Formula\ (3.4)$$

Where (N) is a (Z) score normalization function that normalizes the result to have a mean 0 and standard deviation 1. And (α) is a weight on (K), the cost function (c_i) are weighted by (w_i) . Each weight ((α) and (w_i)) express the relative importance of each term of the sum in the performance of the system.

Here, (N) is used to overcome the problem that the values of (c_i) are not on the same scale as (K) and that the cost measures (c_i) may also be calculated over widely varying scales (e.g. response delay could be measured using seconds while, costs were calculated in terms of NUU). This problem was solved by normalizing each factor (x) to its (Z) score. The equation for normalization is expressed by Carletta as:

$$N(x) = \left(\frac{x - \bar{x}}{\sigma}\right) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots Formula\ (3.5)$$

Where (α_x) is the standard deviation for (x).

To compute the overall performance as in the goal of the study, we obtained the average (c_i), taking note of equation (3.4).

3.4.7 Evaluating the aspect of advancing conversation independently

PARADISE evaluation considered the aspect of advancing conversation as one aspect that contributes to the overall performance. Therefore, the score on performance is inclusive of the aspect. However, due to its significance to the study, it was necessary to measure this aspect independently. To independently evaluate the aspect of advancing conversation, the study used the conversational depth. The metric known as coherence was applied to measure responses as the conversational depth deepened. Coherence is usually measured at turn level. In dialog systems conversations, there is the possibility of context to be carried over multiple turns. The interaction in Mshauri-Wako Chatbot is an example of a multi-turn conversation. To evaluate the Mshauri-Wako Chatbot on conversational depth, we used the total conversation-turns and a topical model to identify the domain for individual utterance. Conversational depth was obtained by averaging the number of consecutive turns (NUU) on the same topic within a domain. Using NUU to compute coherence to give the measure.

Coherence

A coherent response indicates a relevant and comprehensible response to a user's request. A response was deemed weakly coherent if it is somewhat related. For example, when a user says: " What do you think about the symptoms in week four of pregnancy? " the response should be about pregnancy symptoms, symptoms around the fourth week of pregnancy more broadly or something related. A response related to pregnancy but not exactly an opinion or something different would be considered weakly coherent.

Coherence is evaluated with respect to issues or misunderstandings that arise as the conversation progresses. To capture coherence, the study annotated all the interactions for incorrect, irrelevant or inappropriate responses caused by the progress of the conversation. Using the annotations, we calculated the response error rate (RER). Cuayahuitl et al. [58], defines RER as:

$$RER = \frac{Numbers\ of\ incoherent\ responses}{Total\ number\ of utterances} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots Formula\ (3.6)$$

$$RER(\%) = \frac{Numbers\ of\ incoherent\ responses}{Total\ number\ of utterances}\ \mathrm{X}\ 100 \ldots\ldots\ldots\ldots\ldots Formula\ (3.7)$$

$$Coherence\ (\%) = 100 - RER(\%) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots Formula\ (3.8)$$

Coherence does not disregard the overall performance, but rather supports it.

3.4.8 GQM Evaluation

The conceptual framework informed the realization of usability objectives. The usability objectives informed the selection of important attributes with reference to the ISO standard on usability. The attributes informed the design of questions in the questionnaire, questions in the questionnaire were applied to GQM evaluation based on the two goals presented for GQM evaluation. Tables 3.5 and 3.6 show how the goals, questions and metrics are presented based on GQM model.

Goal 1	Purpose	Implement a DS that support
	Issue	Logically progressing
	Object	Conversation
	Viewpoint	From the user's viewpoint
Question	Q1	Is the DS advancing a conversation?
Metrics	M1	-Support of Sub-dialog to feed into main dialog
	M2	-Occurrence of progressive exchange
	M3	-Number of correct responses
Question	Q2	Are user satisfied?
Metrics	M4	-% Ease of interaction
	M5	-% Enjoyability of interaction
Question	Q3	Is the architecture suitable for advancing conversation?
Metrics	M6	-Realization of conversation goal
	M7	-Naturalness of conversation

Table 3.5: A customized GQM description for goal one.

Goal 2	Purpose	Verify if the
	Issue	DS informatively handles the
	Object	conversation from
	Viewpoint	the user's viewpoint
Question	Q1	Is the exchange relevant to a user query?
Metrics	M1	Classification of the exchanges
	M2	User perception of the conversation
	M3	Number of correct responses
Question	Q2	Does the exchange elicit more information about the query?
Metrics	M4	User willingness to use system again

Table 3.6: A customized GQM description for goal two.

3.5 Data collection and analysis

Data was generated from responses from the questionnaire and the system logs, after running the prototype for at least four attempts. The data obtained was then coded and a confusion matrix created. There after linear regression was conducted with reference to Goal Question Metric [53], and PARADISE [54] approaches. See appendices 1 to 3.

3.6 Apparatus

The study used windows 8 / 10 operating system, dialogflow development kit, python language, google and Facebook deployment platform.

3.7 Discussion

During the build phase, each agent was constructed independently. The agents' initial behavioural attributes were defined. The behavioural attributes included the agent's intents, contexts and entities. An important assumption was made, under this assumption, the agent which was loaded in the environment where another one was situated, assumed the name the loaded agent also referred to as the dialog agent, while the original agent assumed the name master agent. To enable the agents, respond to obligations from the dialog manger and avoid conflict, context in master agent was defined whereas context on the loaded agent was suppressed.

In reference to the platform tool, the choice of dialogflow was informed by considering a number of essential factors. Namely: Support for agency, adequacy of libraries for reinforcement learning, ability to integrate a knowledge base and other resources, and support for deployment. However, four realities were encountered, there is no complete toolkit that supports both multi-agency and reinforcement learning. NLU and the dialog manager is integrated with the toolkit. Toolkits supporting agency and ML are in testing, therefore not available for use. Lastly, there was no complete open source toolkit. The study overcame these shortcomings through customization of required functionalities.

In reference to PARADISE, a performance measure is a function of both task success (K) and dialogue costs (c_i). It allows us to evaluate performance at any level of a dialogue, since (K) and (c_i) can be calculated for any dialogue subtask. While it is possible to measure performance over any sub-task, the interest of the study, was to measure performance for the whole dialogue. Therefore, (c_i) for the entire dialog was calculated, through computing the average (c_i).

CHAPTER FOUR

EXPERIMENTATION RESULTS AND DISCUSSIONS

In this chapter, the results of the study are presented and discussed with reference to the aim of the study. The two sub aims – the first to present the results from execution of the artifact, and the second to evaluate the aspect of interest in the study using global acceptable baselines and or benchmarks – form the main justification of the validity of the results. The set-up of the experiment and test procedure is presented and the results from the experiment discussed. In the subsequent section, the details on evaluation of key aspects in reference to the aim of the study is presented and the evaluation results discussed. The chapter concludes with overall discussion within the confines of the evaluation paradigms explored.

4.1 Results from Execution

Mshauri-Wako, the prototype in the study, is an example of an intelligent information retrieval system designed to accomplish some task within the maternal healthcare domain. It is task-oriented, which implies that it's designed to enable the achievement of some tasks. Within the context of this study advancing the conversation is not an end in itself but a means to an end. The advancing conversation is a necessary aspect required to achieve diagnosis and recommendation or advice within the maternal domain.

The desired advancement in the conversation should be meaningful to enable the system to accomplish its goal within the confines of a domain of application. The baseline for identifying advancement in conversation lies in the ability to accomplish a goal. The defined goal is to offer advice and respond to user queries regarding pregnancy. In this set up the thesis confines the concept of advancing the conversation to the ability to continuously refer to the previous statement in the current query to logically solicit information without losing the context and with the view to narrow down a search and offer advice.

An excerpt showing advancing conversation in a dialogue between a user and the system is presented. See figure 4.1.

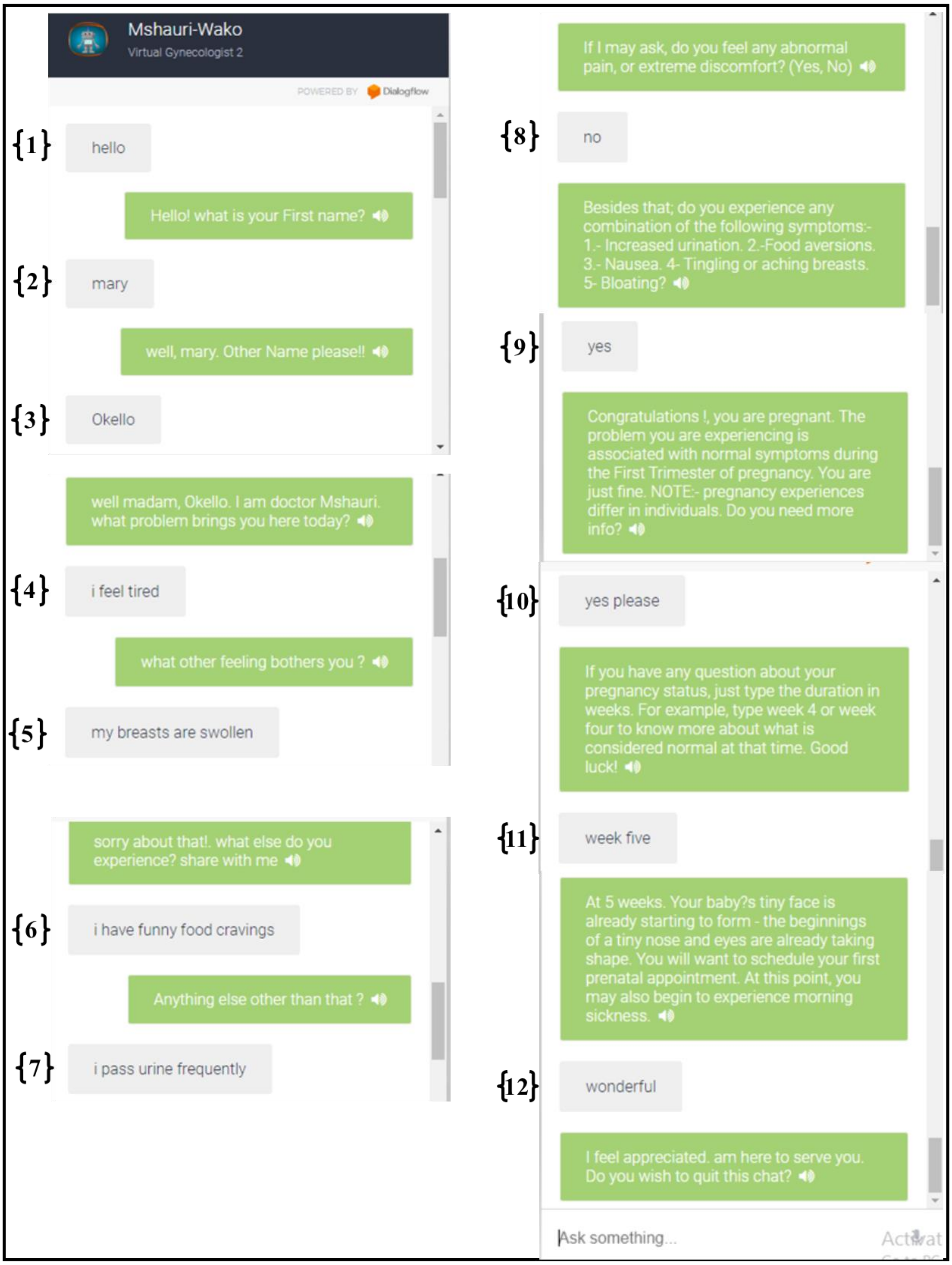


Figure 4.1. A human-agent dialogue during the process of Information inquiry. The dialogue consists of 12 turns. Turns 4 to 9 show the progressive gathering of information which is then associated to some outcome in turn 9. Turns 10 and 11 show additional information on the subject. Turn 12 show successful task-completion.

The study presents a comparative conversation showing ambiguity in identifying whether a conversation advances or not. In other word it is not possible to objectively tell if there was advancement or not; a phenomenon that describes the dilemma for most CODS. If such a progression does not lead to achievement of a goal, then it can be argued that there is no logical advancement in such a conversation. See appendix 8 figure A8.1.

In figure A8.1. the conversation progresses well, however, it seems quite ambiguous to identify "advancement in the conversation". In this case, it is not possible to tell whether there was the advancement or not; - a reality for most CODS. Based on structural design it is rather cumbersome for humans to confirm or deny the possibility of "advancing conversation" in CODS.

The following conversation diagrams show selected distinct conversations from the prototype. These Human-Chatbot conversation classifies the conversation into four distinct activities; salutation, diagnosis, conclusion and general. See figures 4.2, 4.3, 4.4, 4.5 and 4.6.

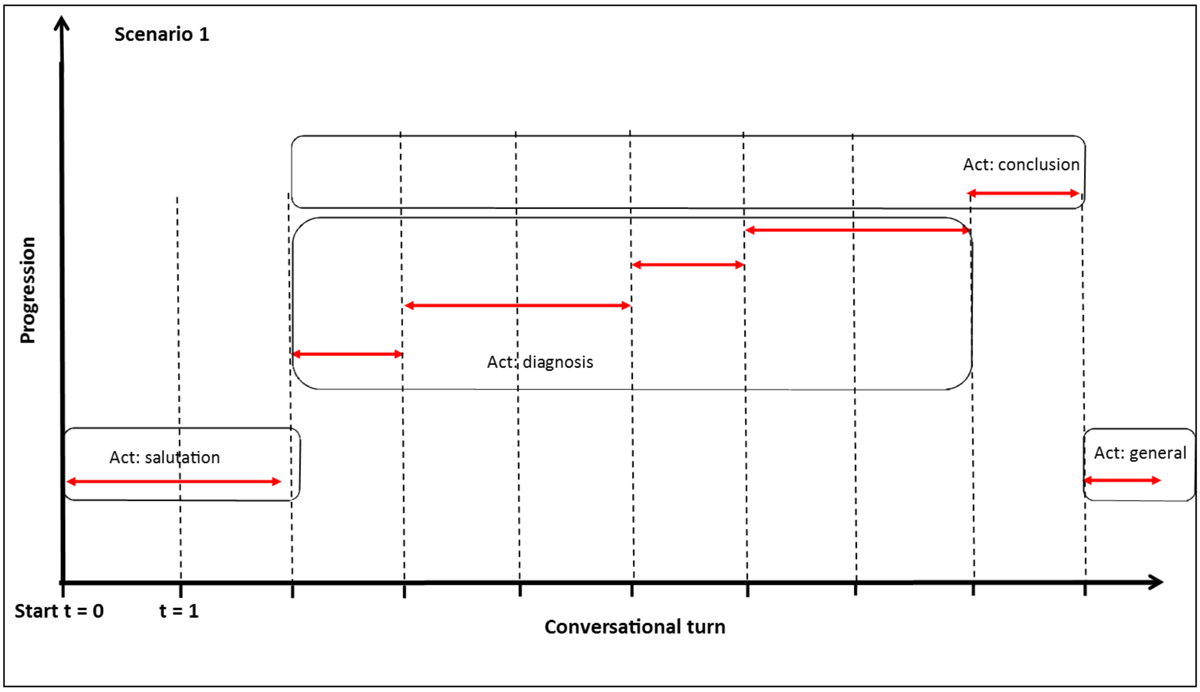


Figure 4.2. The diagnosis activity takes six turns; the arrows are not on the same level, this shows evidence of progression which occurs in both single and multi-turn. Meanwhile, salutation and general activities do show some defined starting context. The overlap between the conclusion and the diagnosis means the conclusion makes reference to the diagnosis.

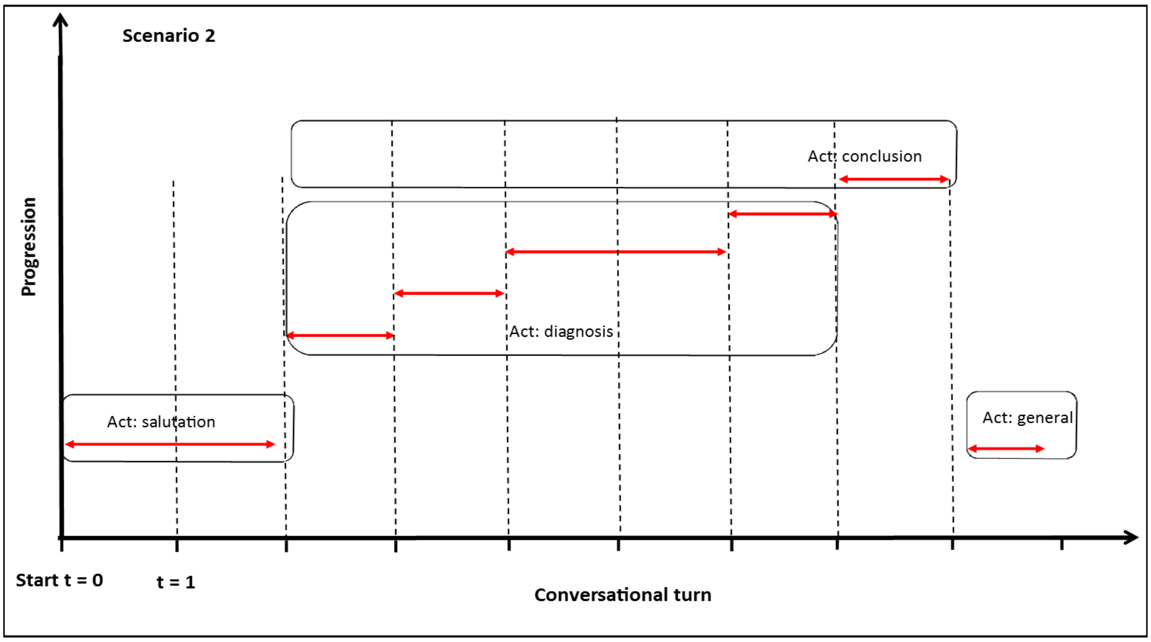


Figure 4.3. In this Human-Chatbot conversation, diagnosis and conclusion activities the arrows are not on the same level, this show progression. Diagnosis activity takes five conversational turns, where progression occurs in both single and multi-turn. On the other hand, salutation and general activities show some defined starting context.

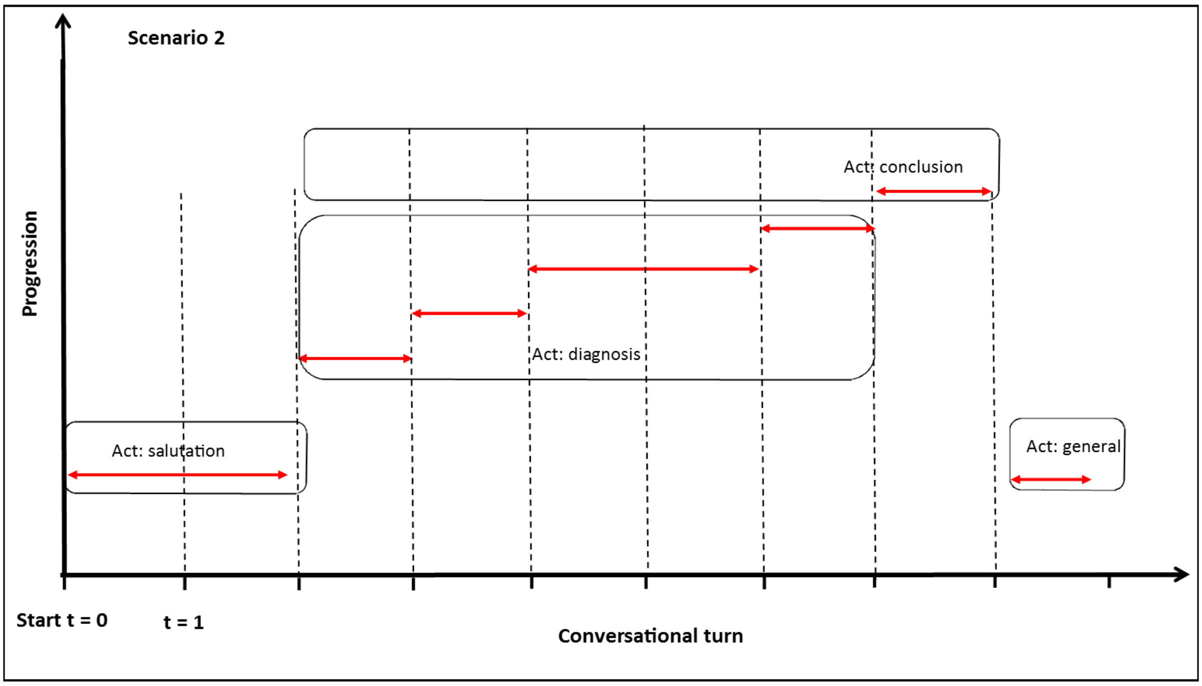


Figure 4.4. In this Human-Chatbot conversation, diagnosis and conclusion activities, the arrows are not on the same level, this show progression. Diagnosis activity takes five conversational turns, where progression occurs in both single and multi-turn. On the other hand, salutation and general activities show some defined starting context.

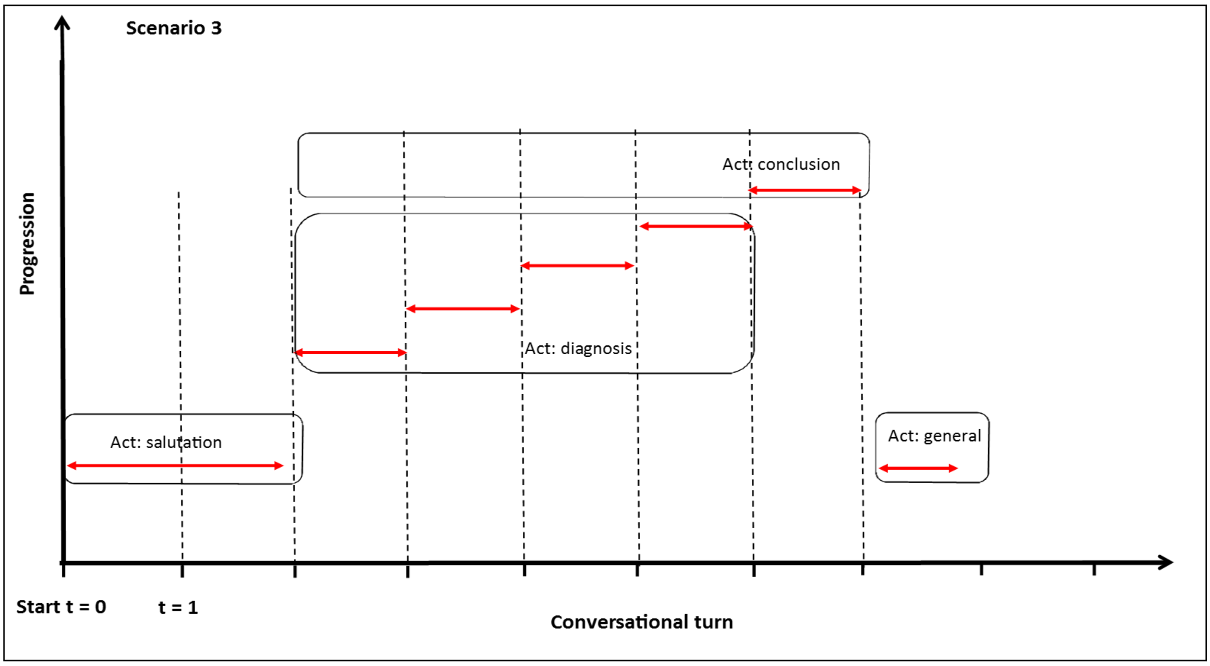


Figure 4.5. In this Human-Chatbot conversation, diagnosis and conclusion activities, the arrows are not on the same level, this show progression. Diagnosis activity takes four conversational turns, where progression occurs in single turn only. On the other hand, salutation and general activities do not show progression.

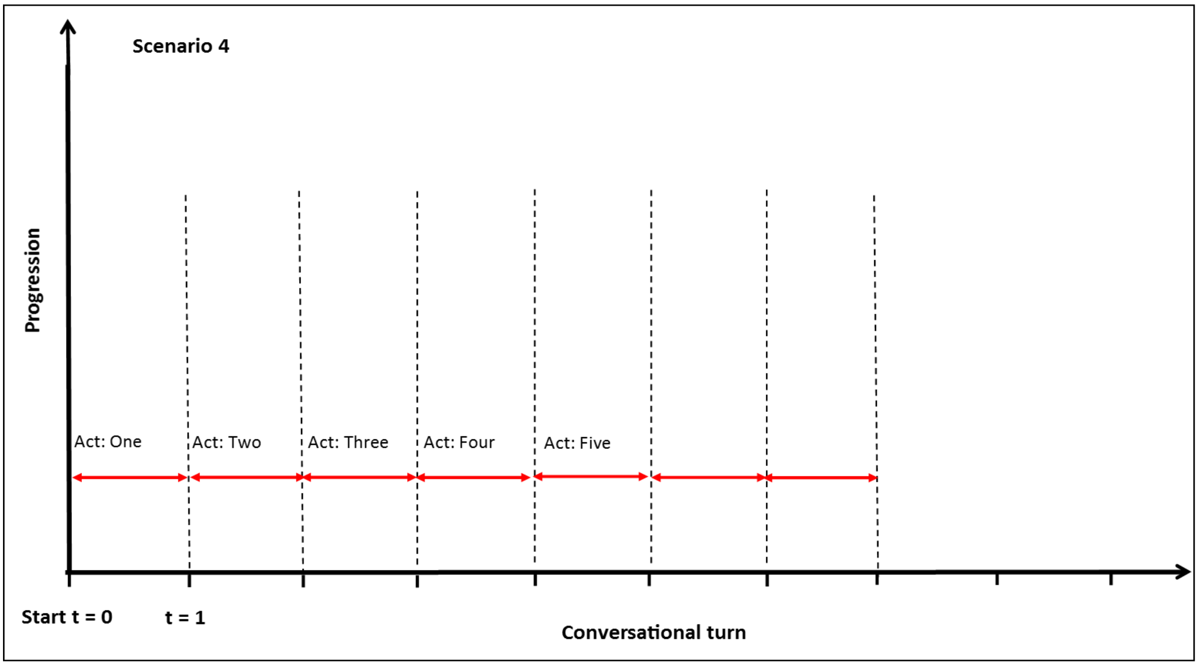


Figure 4.6. In this Human-Chatbot conversation, the activities one to five occur at different conversational turn, the arrows are on the same level, this show that there is no progression. Each activity seems independent occurring at some "starting" context.

This scenario in figure 4.7 depicts a ten turn conversation between the user and the chatbot. This conversation portrays a departure in the understanding. The chatbot and the human seem not to understand each other thus the achievement of a task is unlikely. Figure 4.8. portrays a scenario where the chatbot and the human seem to understand each other thus, achieving a task is a likely goal.

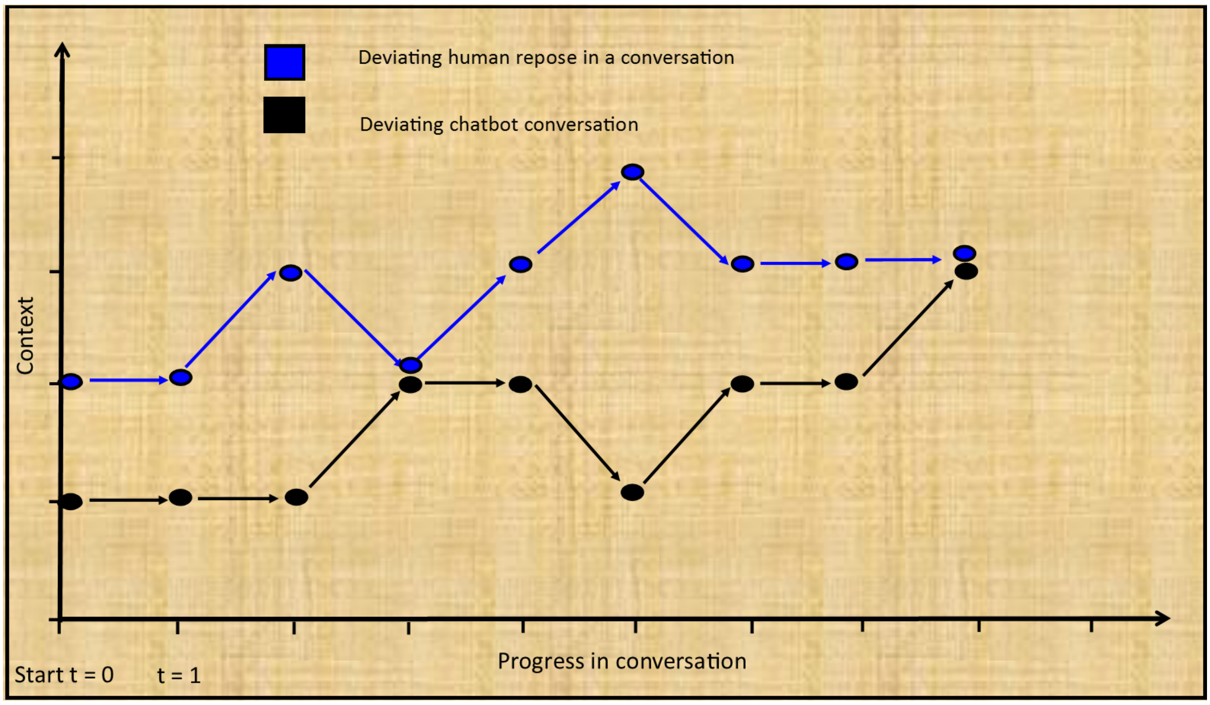


Figure 4.7. A human-agent dialogue during the process of Information inquiry. Shows the human did not understand what the chatbot required. As such the 8 - turn dialog, does not lead to any task achievement, hence a fail in advancing conversation.

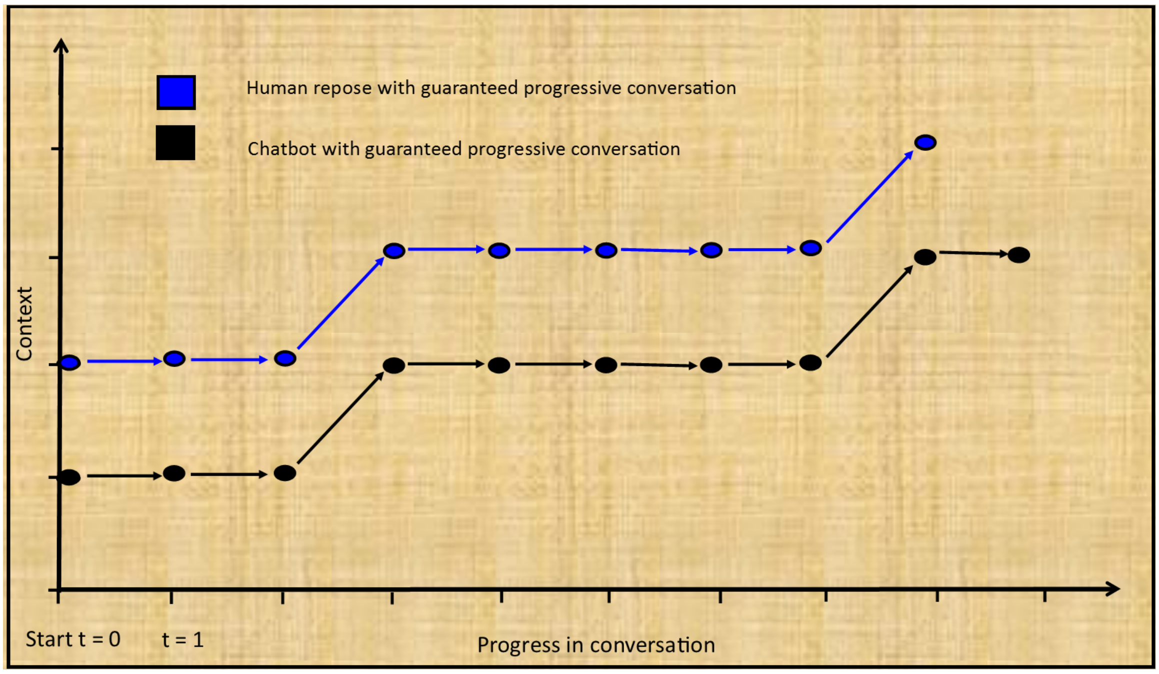


Figure 4.8. A human-agent dialogue during the process of Information inquiry. Shows the human understood what the chatbot required. As such the 8 - turn dialog leads to task achievement, hence a pass in advancing conversation.

Figures 4.9 and 4.10 shows two chatbot conversation, the broken arrow shows a defined "starting context", every conversational turn refer to this "starting context". The continuous arrow show context changing at some conversational turns, thus grouping the context as A, B and C respectively.

The difference between Figure 4.9 and 4.10, lies in the conversational turns in the contexts; - that is, there are different conversational turns for the context A, B and C.

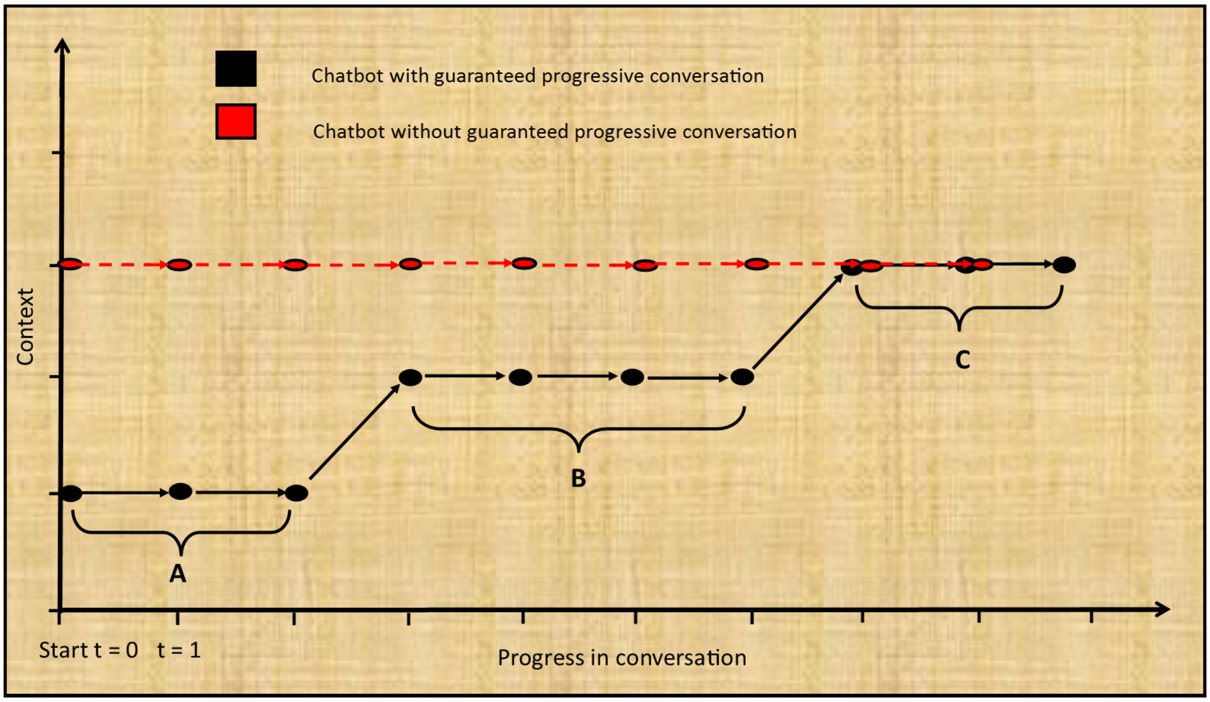


Figure 4.9. Chatbot-Chatbot conversation, showing the behaviour of the conversation context as the conversational turn progresses. Continuous arrow shows change in context after at least one turn while dotted arrow shows no change in the context.

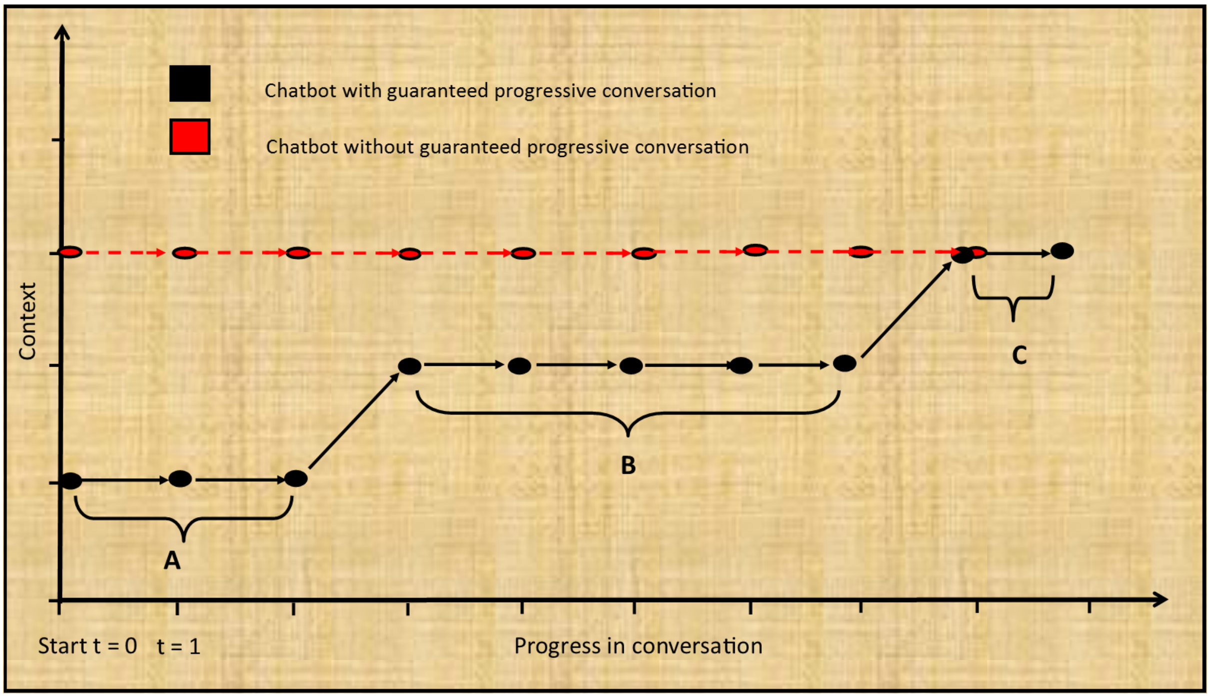


Figure 4.10. Chatbot-Chatbot conversation, showing the behaviour of the conversation context as the conversational turn progresses. Continuous arrow shows change in context after at least one turn while dotted arrow shows no change in the context. A, B and C varying turns within a context.

4.2 Experimentation Results

The study categorized attributes in line with the concept of usability, then evaluated the extent to which the Chatbot fulfilled the specific attribute categories. The categories were functionality, accessibility, affect, and humanity. Participants filled an online questionnaire customized with respect to ISO 9241 concept of usability as discussed in Abran et al. [55]. The following specific responses were drawn from the survey filled in the evaluation process.

4.2.1 Functionality Aspects

Functionality is an important contributor to overall performance and usability. It is essential towards the paradise evaluation. The responses in chart 4.1 to 4.3 were inclined to the functionality aspect. The objective of chart 4.1. was to know how efficient the Chatbot was in accomplishing a given user's task. The responses were measured on five-point Likert scale having items Never, Rarely, Sometimes (neither often nor rarely), Often and Always.

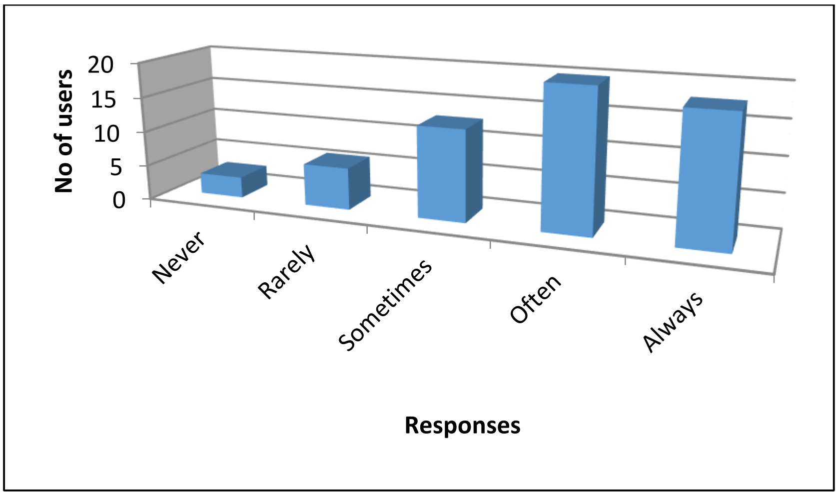


Chart 4.1. Response on whether the system was effective in accomplishing a user's task.

Chart 4.1, indicates how convinced the respondents felt on whether the Chatbot was effective with respect to accomplishing a task. Of the total 60 respondents, 38 respondents indicated strongly that the Chatbot guaranteed task completion, 13 respondents were not convinced on guaranteed task completion while 9 respondents felt there was no guarantee on task completion. Expressed as percentage, 63.33% viewed the system as effective in accomplishing a user's task. 21.67% of the respondents were not fully convinced of the

Chatbot ability to always accomplish a task. Only 15% viewed the Chatbot as not effective in task accomplishment. See appendix 1 for details.

The objective of Chart 4.2. was to know whether the Chatbot could solicit information from a user in a logical manner. The responses were measured on five-point Likert scale having items Never, Rarely, sometimes (neither often nor rarely), Often and Always.

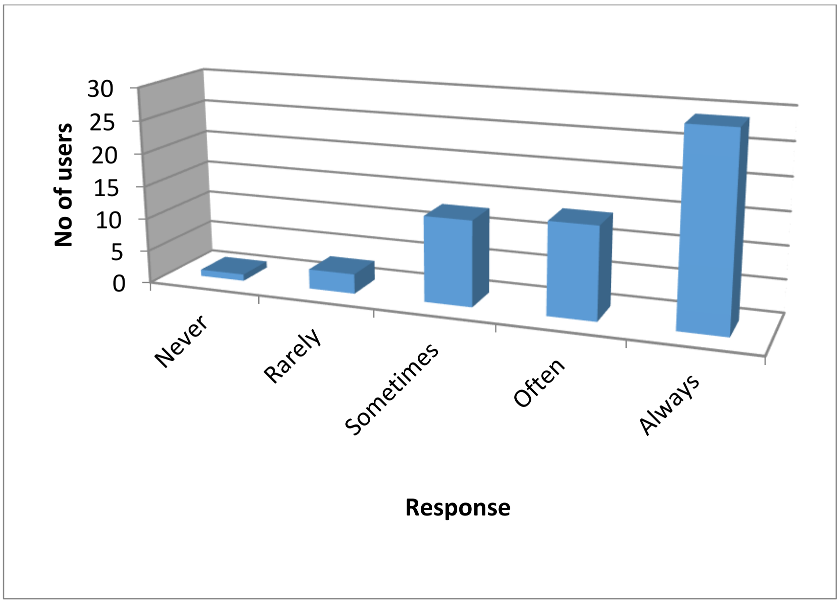


Chart 4.2. Chatbot ability to solicit information in a logic manner.

Chart 4.2 indicates that of the total respondents, 43 respondents strongly approved that the information soliciting process of the Chatbot as logical, 13 respondents could not indicate whether the information soliciting process was logical or not while 4 respondents felt there was no logic in the information soliciting process. Expressed as percentage, 71.66% viewed the information soliciting process as logical. 21.67% did neither agree nor refute that the information soliciting process was logical. Only 6.67% of the respondents viewed the information soliciting process as not logical.

The objective of Chart 4.3. was to know whether the Chatbot was able to maintain the theme of the discussion. It was measured on five-point Likert scale having items Never, Rarely, Sometimes (neither often nor rarely), Often and Always.

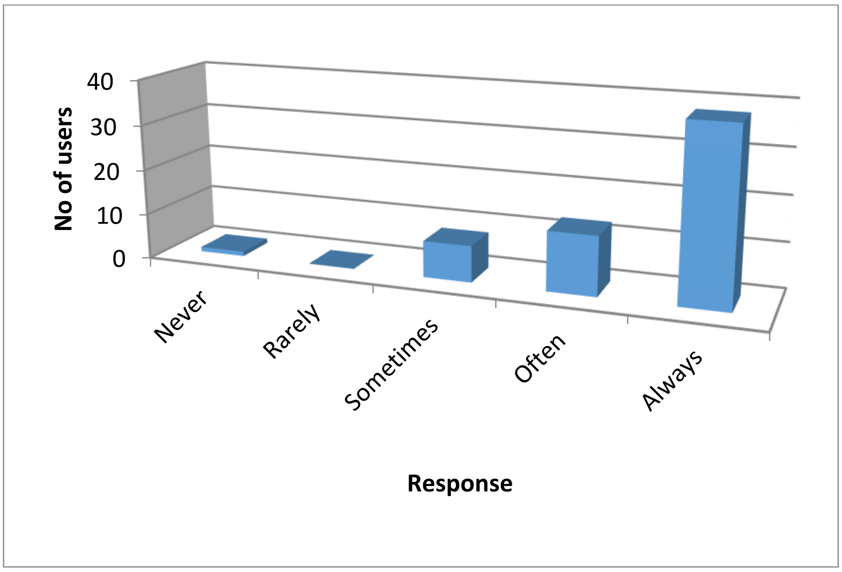


Chart 4.3. Chatbot ability to maintain the theme of the discussion.

Chart 4.3 indicates how the respondents viewed the Chatbot ability to maintain the theme of the discussion. Of the total 60 respondents, 51 respondents were certain that the theme of the discussion was always maintained during an interaction, 8 respondents were not so sure that the Chatbot always maintained the theme, while 1 respondent felt the Chatbot did not maintain the theme of discussion during an interaction. Expressed as percentage, 85% viewed the Chatbot as able to maintain the theme of the discussion. 13.33% did neither agree nor refute that the Chatbot was able to maintain the discussion theme. Only 1.67% of the respondents viewed the Chatbot as not able to maintain the theme of discussion.

4.2.2 Accessibility Aspects

The responses in chart 4.4 to 4.6 were inclined to the accessibility aspect. The objective of chart 4.4. was to know whether the Chatbot was able to detect intent or meaning during an interaction. It was measured on five-point Likert scale having items Never, Rarely, Sometimes (neither often nor rarely), Often and Always.

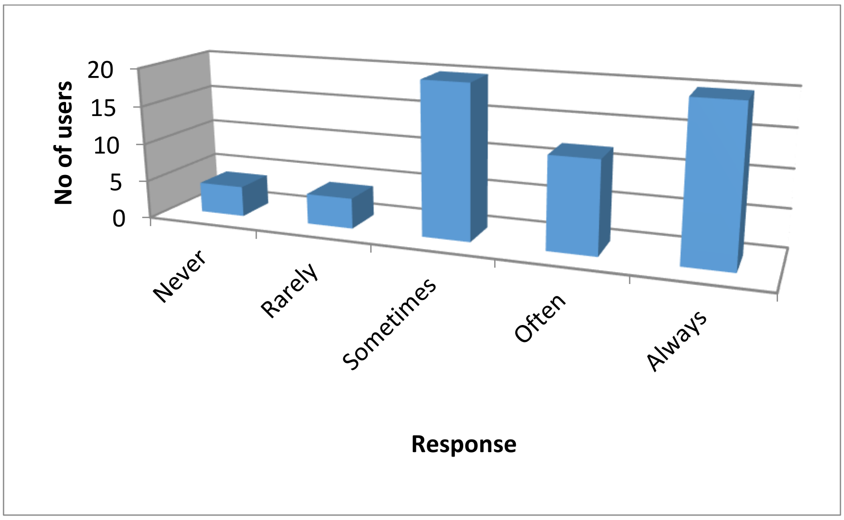


Chart 4.4. Chatbot ability to detect intent or meaning.

Chart 4.4 indicates how the respondents perceived the Chatbot ability to detect intent or meaning during an interaction. Of the total 60 respondents, 32 respondents were confident that the Chatbot was always able to detect intent or meaning during an interaction, 20 respondents were not so sure of the Chatbot ability to always detect intent or meaning during an interaction, while 8 respondents felt the Chatbot did not detect or possessed the ability to detect any intent or meaning during an interaction. Expressed as a percentage; 53.33% viewed the Chatbot as able to detect intent and meaning during an interaction. 33.33% did neither agree nor refute that the Chatbot ability to detect intent or meaning during an interaction. Only 13.34% of the respondents viewed the Chatbot as not able to detect either intent or meaning during an interaction.

The objective of Chart 4.5. was to know whether the Chatbot made the conversation any easy. It was measured on five-point Likert scale having items Never, Rarely, Sometimes (neither often nor rarely), Often and Always.

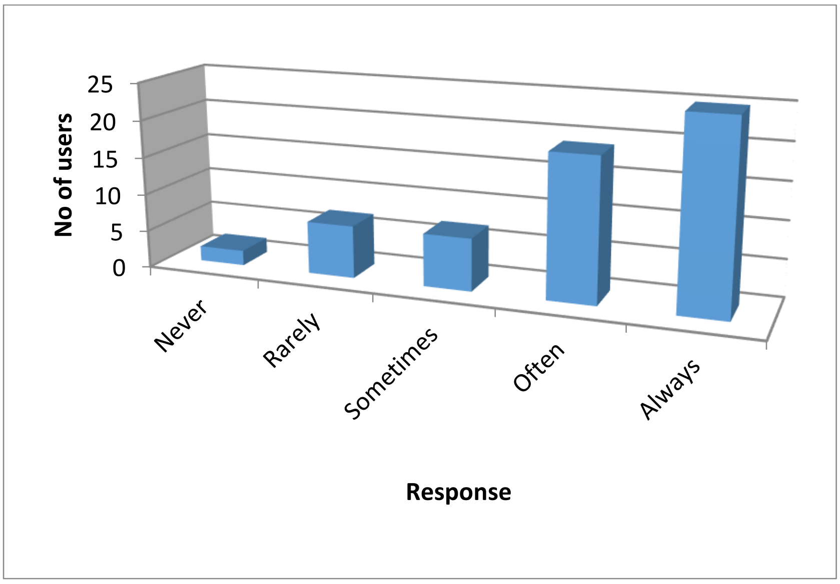


Chart 4.5. The Chatbot ability to make the conversation easier.

Chart 4.5 indicates how the respondents perceived the Chatbot ability to make the conversation easier. Of the total 60 respondents, 44 respondents were confident that the Chatbot made the conversation easy, 7 respondents were not so sure on whether the conversation was easy or not, while 9 respondents felt the Chatbot did not make the conversation any easy. Expressed as a percentage, 73.34% viewed the Chatbot conversation as easy. 11.67% could not tell whether the conversation was easy or not. Only 15% of the respondents viewed the Chatbot conversation as not easy.

The objective of Chart 4.6. was to know whether the Chatbot provided needed information easily. It was measured on five-point Likert scale having items Never, Rarely, Sometimes (neither often nor rarely), Often and Always.

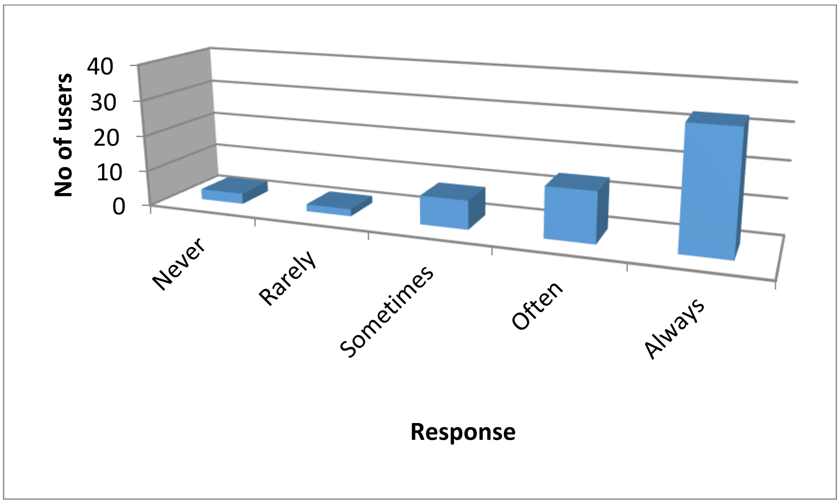


Chart 4.6. Chatbot ability to make provide needed information easily.

Chart 4.6 indicates how the respondents perceived the Chatbot ability to make provide needed information easily. Of the total 60 respondents, 47 respondents were certain that the Chatbot provided needed information easily, 8 respondents were not so sure, while 5 respondents felt the Chatbot did not easily provide needed information. Expressed as a percentage, 78.33% were certain that the Chatbot provided needed information easily. 13.33% of the respondents were quite uncertain. Only 8.33% of the respondents felt that the Chatbot did not provide the needed information easily.

4.2.3 Affect Aspects

The responses in chart 4.7 to 4.9 were inclined to the affect aspect. The objective of chart 4.7. was to know whether the Chatbot maintained a natural satisfying interaction. It was measured on five-point Likert scale having items Never, Rarely, Sometimes (neither often nor rarely), Often and Always.

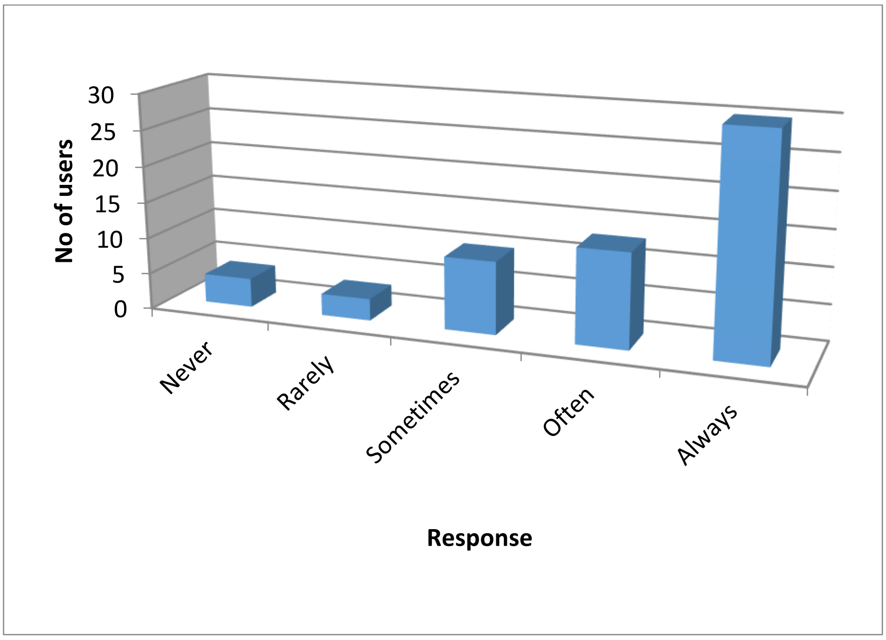


Chart 4.7. Chatbot ability to maintain a natural satisfying interaction.

Chart 4.7 indicates how the respondents perceived the Chatbot ability to maintain a natural satisfying interaction. Of the total 60 respondents, 43 respondents were certain that the interaction was natural and satisfying, 10 respondents were not so sure, while 7 respondents felt the interactions with the Chatbot were not natural and satisfying. Expressed as a percentage, 71.67% were certain that the interaction was natural and satisfying. 16.67% of the respondents were uncertain. Only 11.67% of the respondents felt that the Chatbot interactions were neither natural nor satisfying.

The objective of Chart 4.8. was to know whether the Chatbot maintained a fluent dialogue. It was measured on five-point Likert scale having items Never, Rarely, Sometimes (neither often nor rarely), Often and Always.

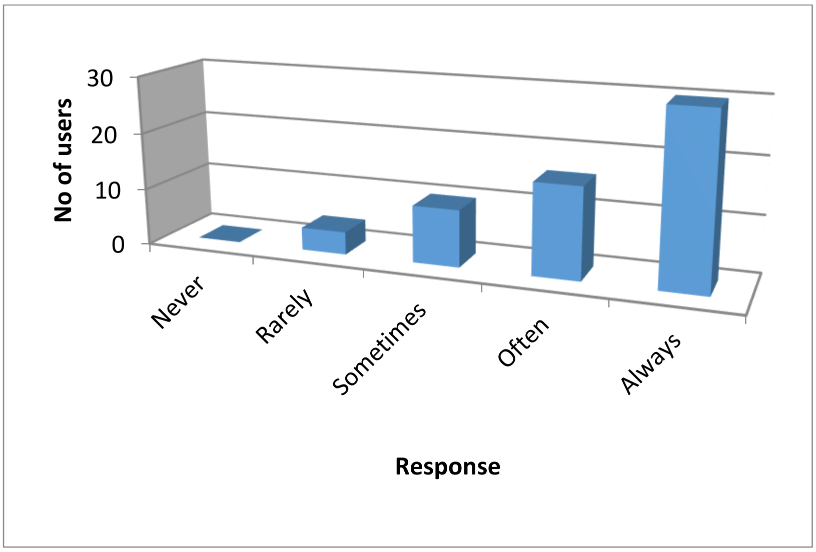


Chart 4.8. Chatbot ability to maintained a fluent dialogue

Chart 4.8 indicates how the respondents perceived the Chatbot ability to maintained a fluent dialogue. Of the total 60 respondents, 46 respondents were certain that the dialogue was fluent at all times, 10 respondents felt the dialogue was fluent a few times, while 4 respondents felt the dialogue was never fluent at all. Expressed as a percentage, 76.67% were certain that the Chatbot had and maintained a fluent dialogue. 16.67% felt the Chatbot maintained a fluent dialogue but not always. Only 6.67% of the respondents felt that the Chatbot did not maintain a fluent dialogue.

The objective of Chart 4.9. was to know whether the Chatbot conveyed personality during an interaction. It was measured on five-point Likert scale having items Never, Rarely, Sometimes (neither often nor rarely), Often and Always.

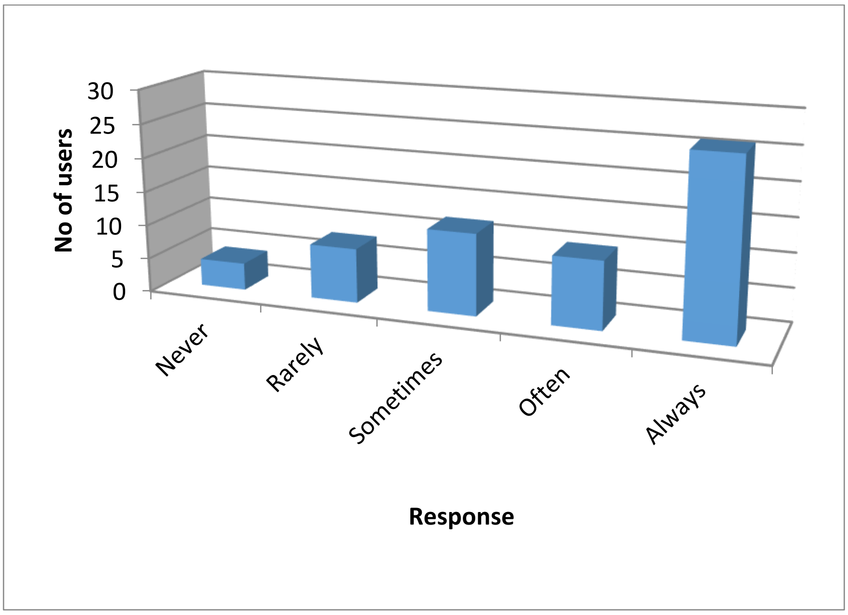


Chart 4.9. Chatbot conveyed personality during interaction

Chart 4.9 indicates how the respondents perceived the Chatbot ability to Chatbot convey personality during an interaction. Of the total 60 respondents, 36 respondents were certain that the Chatbot conveyed personality during an interaction, 12 respondents felt the Chatbot conveyed personality only a few times while 12 respondents felt there was no personality in the Chatbot conversation at all. Expressed as a percentage, 60% were certain that the Chatbot conveyed personality during an interaction. 20% felt the Chatbot conveyed personality only a few times. Only 20% of the respondents felt the Chatbot did not convey any personality during an interaction.

4.2.4 Humanity Aspects

The responses in chart 4.10 to 4.12 were inclined to the humanity aspect. The objective of chart 4.10 was to know how easy it is to use the Chatbot. It was measured on five-point Likert scale having items Never, Rarely, Sometimes (neither often nor rarely), Often and Always.

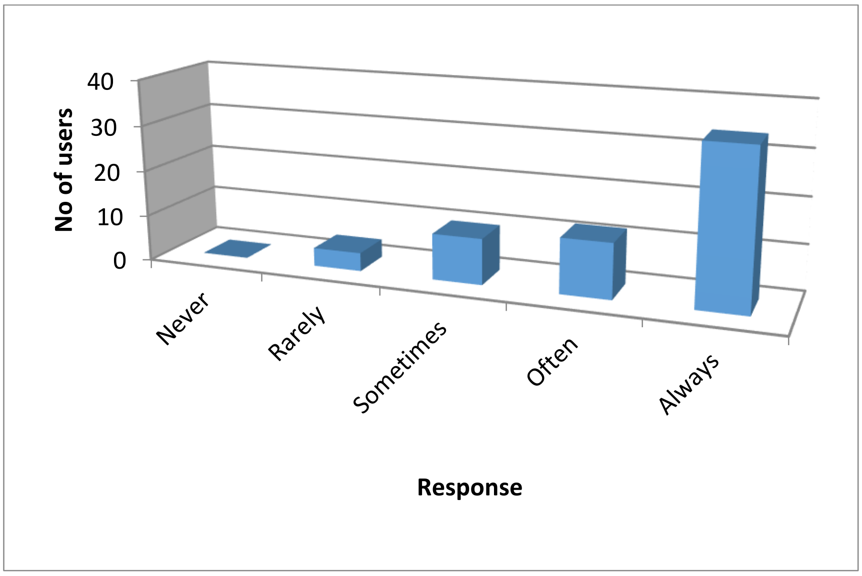


Chart 4.10. How easy is it to use the Chatbot.

Chart 4.10 indicates how the respondents perceived how easy it was to use the Chatbot during interactions. Of the total 60 respondents, 45 respondents perceived the Chatbot as very easy to use, 10 respondents, perceived it as neither easy nor hard, while 5 respondents perceived it as not easy to use. Expressed as a percentage, 75% viewed the Chatbot as very easy to use. 16.67% viewed the Chatbot neither easy nor hard to use. Only 8.34% of the respondents viewed the Chatbot as not easy to use.

The objective of Chart 4.11. was to know whether users enjoyed using the Chatbot. It is measured on five-point Likert scale having items Never, Rarely, Sometimes (neither often nor rarely), Often and Always.

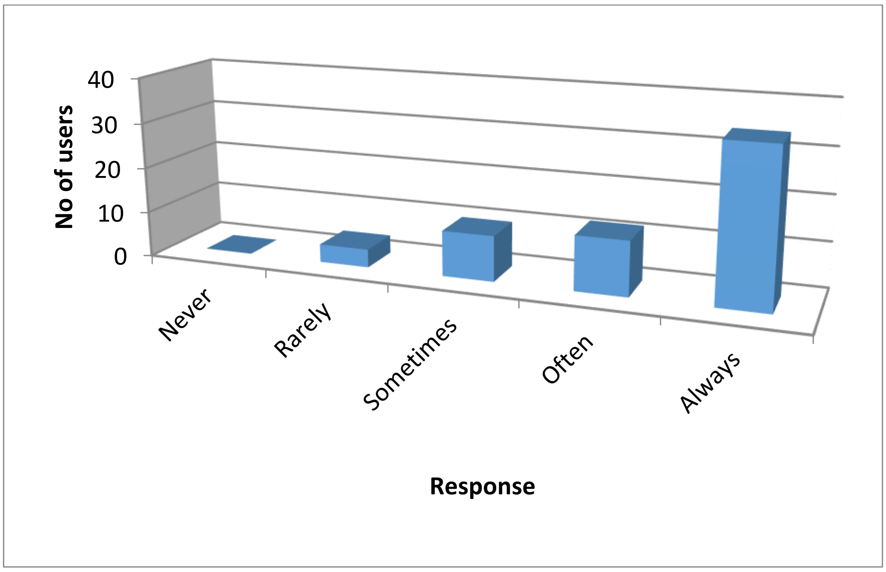


Chart 4.11. User's enjoyed the time spent interacting with the Chatbot.

Chart 4.11 indicates how the respondents enjoyed using the Chatbot during interactions. Of the total 60 respondents, 46 respondents enjoyed the interactions, 10 respondents were not sure whether they enjoyed or not, while 4 respondents did not enjoy the interactions. Expressed as a percentage, 76% enjoyed using the Chatbot. 16.67% partly enjoyed using the Chatbot. Only 6.64% did not enjoy using the Chatbot.

The objective of Chart 4.12. was to know whether users will want to use the Chatbot again. It was measured on five-point Likert scale having items Never, Rarely, Sometimes (neither often nor rarely), Often and Always.

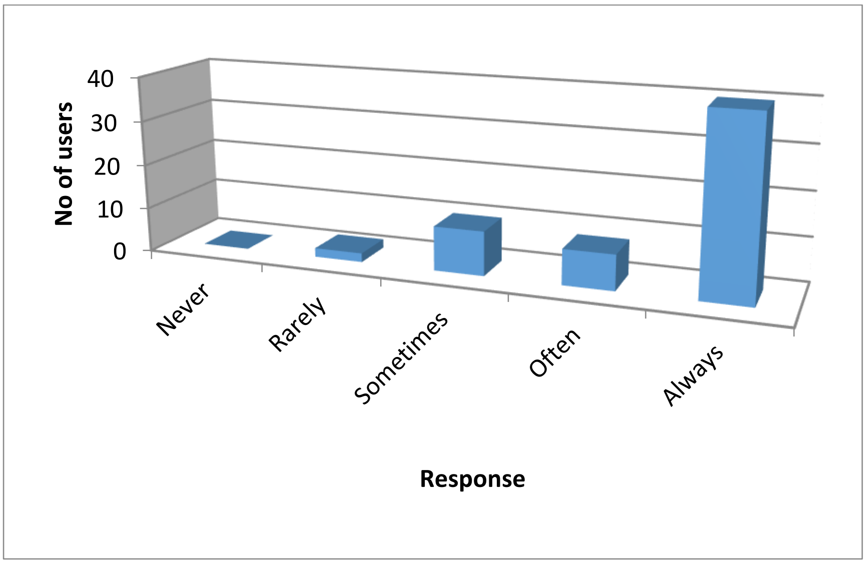


Chart 4.12. the respondent's willingness to use the Chatbot again

Chart 4.12 indicates the respondent's willingness to use the Chatbot again. Of the total 60 respondents, 48 respondents were very willing to use the Chatbot again, 10 respondents were not sure, while 2 respondents were not willing to use the Chatbot again. Expressed as a percentage, 80% would use the Chatbot again. 16.67% were not sure about using the Chatbot again. Only 3.33% did not want to use the Chatbot again.

The responses as presented in tables 4.1 to 4.12 were classified into four main categories namely functionality (*FX*), humanity (*H*), affect (*AF*) and accessibility (*AC*), in reference to the ISO 9241 standards. Each attribute category was coded and analysed to make the data usable in measuring different quantities within the paradise model.

User satisfaction was derived from the evaluation questionnaire where a summary report based on average scores on a Likert scale was used. Highly satisfied was awarded 5 points while partly satisfied was awarded 1 point. The number of turns *(NUU)* was obtained from the analytics engine logs during each user interaction. The computed (k) was used for all the respondents. See appendix 2, Table A3 show the coded data with the number of turns *logs while* Table B3 show complete data with summarized scores.

4.3 Measuring Task Success

The user data presented in table A1 and AVM scenario keys, presented in table 3.5 were used to construct a confusion matrix, M, as presented in table 4.1.

N = 60		Greetings				Names				User Problem				System Response				More Information			
	DATA	V1	V2	V3	V4	V1	V2	V3	V4	V1	V2	V3	V4	V1	V2	V3	V4	V1	V2	V3	V4
	V1	12			1																
	V2		26		1																
	V3			10	1																
Greetings	V4			1	8																
	V1					4															
	V2						10														
	V3							13													
Names	V4						1		32												
	V1									3											
	V2										13										
User	V3									1		14									
Problem	V4												29								
	V1													3	2						
	V2														2						
System	V3													1		33					
Response	V4													2	2		14				
	V1																1	3		1	
	V2																		10		1
More	V3																	1		30	
Information	V4																			1	13
SUM		12	26	11	11	4	11	13	32	4	13	14	29	6	6	33	15	4	10	32	14

Table 4.1: The confusion matrix M, illustrates the number of times the system behaved correctly.

The matrix, M in table 4.1 was generated in an evaluation of 60 complete dialogues. Complete dialogues were identified based on the number of turns and user goal achievement. Labels v1 to v4 represent the possible values for categories greetings, names, user problem, system response, and more information, respectively in each matrix. Columns represent the key, specifying the information values the agent and user were supposed to communicate to one another given a particular scenario. The blanks in columns suggest no offer guidance was given on further response. The values in the cells of the matrix are based on comparisons between the dialogue and scenario key AVMs. Whenever an attribute value in a dialogue (i.e., data) AVM matches the value in its scenario key, the number in the appropriate diagonal cell of the matrix is incremented by 1. The off diagonal cells represent the misunderstandings that are not corrected in the dialogue.

Using the matrix M, it was possible to measure the task success for a sub-dialogue and for whole dialogue. The focus in the study was to measure task success for a whole dialogue.

This involved computing the Kappa coefficient (K), from the matrix M. Then computing $P(E)$ and $P(A)$ respectively.

Equation (3.2) was applied to obtain a $P(E)$ of 0.061. Equation (3.3) was applied to obtain a $P(A)$ of 0.940. Lastly, equation (3.1) was applied to yield a (K) of 0.936. as presented in table 4.2.

Formula	Score
$Formula\ (2)..P(E) = \sum_{i=1}^{n} \left(\frac{t_i}{T}\right)^2$	0.061
$Formula\ (3)..P(A) = \sum_{i=1}^{n} \left(\frac{M(i,i)}{T}\right)$	0.940
$Formula\ (1)..K = \frac{P(A) - P(E)}{1 - P(E)}$	0.936

Table 4.2: Summary results for task success.

4.4 Estimating the overall system performance

Since the overall performance was of interest to the study, the AVM attributes namely *AC, AF, FX, H* and *NUU* were tagged with respective costs. Then a performance function was applied for the overall performance measure. A summary of attributed with associated cost as presented in Table 4.3.

Attributes	σ_x	$\bar{x}$
AC	0.929127	4.533333
AF	1.290009	3.7833333
FX	1.026623	3.883333
H	0.971195	4.35
NUU	9.501769	22.567

Table 4.3: Summary of attributes with associated costs.

This required however, that values be on the same scale as (K). The attribute *NUU* which qualified as our (c_i) was in a different scale, therefore, Equation (3.5) was applied for normalization. (see appendix 2 for full table). The average (c_i) was computed and the result is normalized, as presented in Table 4.4.

(c_i) Averaged value	(c_i) Averaged to nearest integer	Z score
22.567	23	0.046

Table 4.4: Normalized $(\boldsymbol{c_i})$ score.

To obtain the overall performance, Equation (3.4) was then applied, however, the equation is not complete since the values for the weights (α) and (w_i) were still unknown. To determine the unknown values, multiple regression analysis was applied to provide the weights. Table 4.5 show the regression statistics.

$$P = (\alpha * \mathrm{N(k)}) - \sum_{i=1}^{n} w_i * N(c_i)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots Equation\ (3.4)$$

ANOVA								
	df	*SS*	*MS*	*F*	*Significance F*			
Regression	4	47.16909135	11.79227284	189.96164	1.7021E-31			
Residual	55	3.414241982	0.062077127					
Total	59	50.58333333						
	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	0.368099608	0.191046125	1.926757784	0.059182695	-0.014765382	0.750964598	-0.014765382	0.750964598
AC	0.279103156	0.065207408	4.2802369	7.52842E-05	0.148424589	0.409781723	0.148424589	0.409781723
AF	0.113522364	0.065793112	1.725444513	0.09006255	-0.018329979	0.245374707	-0.018329979	0.245374707
FX	0.102290652	0.069840012	1.464642537	0.148711504	-0.037671859	0.242253164	-0.037671859	0.242253164
H	0.449787994	0.095182471	4.725533916	1.63187E-05	0.25903806	0.640537928	0.25903806	0.640537928

Table 4.5: First regression Output, effects of four attributes to user satisfaction

The standard upper bound for calling a result statistically significant is **p < .05,** Cohen [76]. The ANOVA demonstrate the effects of each attribute to user satisfaction (US). The probability **P < .03** indicate that the attributes contribution is statistically significant. The results in table 4.6 shows the overall contribution of our attributes is statistically significant. However individual contribution shows *(FX)* has a p value of .148, hence not statistically significant to explain user satisfaction. For this reason, the attribute *(FX)* was excluded and a second regression analysis performed, to obtain the results shown in table 4.6.

ANOVA								
	df	*SS*	*MS*	*F*	*Significance F*			
Regression	3	48.98495182	16.32831727	572.0697826	5.83308E-42			
Residual	56	1.598381518	0.028542527					
Total	59	50.58333333						
	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	0.724563694	0.116072494	6.24233762	6.11532E-08	0.492042549	0.95708484	0.492042549	0.95708484
AC	0.28649978	0.048938333	5.854301993	2.62045E-07	0.188464519	0.384535042	0.188464519	0.384535042
AF	0.173034462	0.042643222	4.0577249	0.000155148	0.087609824	0.2584591	0.087609824	0.2584591
H	0.405571519	0.066380031	6.109842299	1.00677E-07	0.272596339	0.5385467	0.272596339	0.5385467

Table 4.6: Second regression Output, effects of significant attributes to user satisfaction

This linear regression produces coefficients or weights describing the relative contribution of predictor factors accounting for the variance in a predicted factor. The coefficients were summed to obtain (w_i) of 0.8651 while the intercept 0.72456 formed our α , with respect to equation 3.4. to obtain $N(c_i)$ =0.046 as presented in figure 4.12.

Having all the unknown values ready, Equation (3.4) is applied to obtain the overall system performance, as shown below.

$\boldsymbol{Applied\ Equation\ 3.4}$

$\boldsymbol{p}$ = (0.72456 * (0.8733*0.936)) - (0. 8651* (0.046))

= 0.552813

p = 0.553

$\boldsymbol{p}$ = 55.3 % (as a percentage)

Figure 4.11: Computing the overall performance.

4.4.1 Evaluating the aspect of advancing conversation independently

The conversational depth was used to independently evaluate the aspect of advancing conversation. Conversational depth was obtained by averaging the number of consecutive turns (NUU) on the same topic within a domain. Using NUU to compute coherence to give the measure. Coherence is evaluated with respect to issues or misunderstandings that arise as the conversation progresses.

To capture coherence, the study annotated all the interactions for incorrect, irrelevant or inappropriate responses caused by the progress of the conversation. Using the annotations, the study calculated the response error rate (RER). To capture coherence, the study annotated all the interactions for irrelevant or inappropriate responses, as presented in table 4.7.

label	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	totals
Irrelevant responses	9	8	12	9	4	7	5	10	5	4	4	2	79
Total no. of turns	60	60	60	60	60	60	60	60	60	60	60	60	720
Relevant responses	51	52	48	51	56	53	55	50	55	56	56	58	641

Table 4.7: Summary of response annotation.

Using the annotations, the study calculated the response error rate (RER) and coherence with respect to equations 3.6, 3.7 and 3.8. The computation to obtain coherence is described below. See figure 4.13.

$Applied\ Equation$ 3.6, 3.7. and 3.8

=(N2/N3) = 0.1097

$RER(\%)$ = 10.97%

$Coherence\ (\%) = 100 - RER$

= 100 -10.97

= 89.03%

Figure 4.12: Results on Coherence.

4.4.2 Results from GQM analysis

This evaluation featured two main goals: Goal 1 was refined into three questions, while Goal 2 refined into two questions. Table 4.8 and 4.9 show the responses.

Goal 1	***Purpose***	***Implement a DS that support***	***Response***
	Issue	***Logically progressing***	
	Object	***Conversation***	
	Viewpoint	***From the user's viewpoint***	
Question	***Q1***	***Is the DS advancing a conversation?***	
Metrics	M1	-Support of Sub-dialog to feed into main dialog	Yes
	M2	-Occurrence of progressive exchange	Yes
	M3	-Number of correct responses	93
Question	***Q2***	***Are user satisfied?***	
Metrics	M4	-% Ease of interaction	75
	M5	-% Enjoyability of interaction	80
Question	***Q3***	***Is the architecture suitable for advancing conversation?***	
Metrics	M6	-Realization of conversation goal	78
	M7	-Naturalness of conversation	72

Table 4.8: Summary results for goal number one.

Goal 2	***Purpose***	***Verify if the***	***Response***
	Issue	***DS informatively handles the***	
	Object	***conversation from***	
	Viewpoint	***the user's viewpoint***	
Question	***Q1***	***Is the exchange relevant to a user query?***	
Metrics	M1	Classification of the exchanges	yes
	M2	User perception of the conversation	Enjoyable
	M3	Number of correct responses	93
Question	***Q2***	***Does the exchange elicit more information about the query?***	
Metrics	M4	User willingness to use system again	80

Table 4.9: Summary results for goal number two.

Table 4.10 show the analysis of responses in GQM analysis.

goal 1	
quantitative	
metric 3	93.6
metric 4	75
metric 5	80
metric 6	78
metric 7	72
Mean	**79.72**

goal 2	
quantitative	
metric 3	93.6
metric 4	80
Mean (%)	**86.8**

Table 4.10: Quantitative analysis results, based on GQM.

Tables 4.10 show a summary of the qualitative responses for each goal.

goal 1	
qualitative	
metric 1	yes
metric 2	yes

goal 2	
qualitative	
metric 1	yes
metric 2	enjoyable

Table 4.11: Qualitative analysis results, based on GQM.

Tables 4.11 show the structure of goals, questions and metrics. Goal one verified that the Chatbot supported logically progressing conversation from the user's perspective. While goal two verified that the Chatbot handled the user's conversation informatively. Table 4.10 show quantitatively that goal 1 was achieved to an extent of 79.72%, while goal 2 was achieved to an extent of 86.8%. Table 4.11 show qualitatively that the respondents reaffirm the system's ability to support sub dialogues and acknowledge the occurrence of progressive exchanges. Besides, table 4.11 show qualitatively that the respondents reaffirm being able to complete tasks and enjoyed the interaction.

4.5 Discussion

The system prompts the user for response in turn 4, and the user responds. From the user's perspective, the user presents symptoms progressively in turns 4 through 7. This is similar to a progressive search. From the system's perspective, the system progressively gathers information, compares to what it already knows in turns 8 and 9 and offer advice, as shown in turn 10. The system further inquires if the user wanted additional information, and depending on the user's preference, the system delivers on the user's expectations. The system then branches out of the conversational context, closes the dialogue having achieved the role of an adviser.

Each time the user provides information, the system has to refer to the previous information. At the end it offers advice within the required context. In this respect the conversation advances inwardly. The inward advancing is indeed logical, and that's how the system achieves its goal as a knowledgeable advisor.

The 12 turns human-agent dialogue in Figure 4.1, shows an Information inquiry process. Beginning from the conversational turns 4 to 9 there is the progressive gathering of information. This progressive gathering is then associated to some outcome in turn 9. That's is the Information gathered bears influence to the outcome.

This resonates with a diagnosis and recommendation scenario where the recommendation relies on the diagnosis. For this to be possible the diagnosis should permit a progressive inquiry that is tied to a subject of interest. The progressive inquiry over a subject is what the study refer to as advancing conversation, even though this is inward advancing. The turns 10 and 11 demonstrate a period where advancing is not required, and thus behaves like a normal search. Here there is a defined starting context where all turns begin from. Turn 12 show successful task-completion.

To ascertain whether a conversation advances or not is quite ambiguous, especially if there is no objective to be achieved. Furthermore, to guarantee whether the required advancement is not by luck or chance becomes a difficult undertaking. For the case of CODS, there is no objective and therefore, it is rather difficult by design to prove or disapprove the presence of advancing conversation. In other words, it is not possible to objectively tell if there is advancement or not; a phenomenon that describes the dilemma for most CODS. The purpose and definition of CODS do not demand them to show or possess this ability. To understand the effect of or presence of advancing conversation, it must lead to task achievement.

The conversation diagrams in figure 4.3, classify the conversation into four distinct activities; salutation, diagnosis, conclusion and general; - within the maternal health domain. It shows six conversational turns for the diagnosis activity, where progression occurs during the entire diagnosis activity. Single turns and multiple turns occur based on the inputs gathered during the diagnosis. While direct or obvious input (symptoms) may lead to a shorter inquiry process, indirect input (symptoms) may prolong the inquiry process. That explains the varying number of turns in the diagnosis activity. Since the conclusion derived relies on the diagnosis; - it demands the inquiry be logical. These explain the overlap between the conclusion and the diagnosis.

In figures 4.3 and 4.4, the diagnosis activity took six and five conversational turns respectively; this implied some inputs were not direct symptoms for the diagnosis. In Figure 4.5, the diagnosis activity took four conversational turns; this meant most inputs were direct symptoms for the diagnosis. In all scenarios, single turn progression implied the input was sufficient to trigger the change of context, while multi-turn progression implied more input was required to trigger the change of context.

In an ordinary search, there is always some "starting context". This starting context serves as the reference for every search. The conversation in figure 4.6 depicts an ordinary search, and regardless of the number of conversational turns, the reference is the defined starting context. Based on the characteristics of advancing conversation, it is misleading to purport that such a conversation can advance however close the responses may be. There are situations where the human may choose not to understand the chatbot in such a case, it may be unlikely to achieve a task. When the human and the chatbot agree or seem to understand each other, the conversation pattern is similar, as in figure 4.8. Otherwise, the conversation pattern is dissimilar, as in figure 4.7.

Considering the conversation taken from two Chatbots, i.e. a chatbot from the study and a comparative chatbot. The broken arrow shows searches that are regarded as new whenever they occur. While the continuous arrow show searches that progress in some context. See Figure 4.9. and 4.10.

To picture how the advancing occurs and to assure that the advancing is not by chance, figure 4.17 illustrates the "context aware" conversation from the prototype of the study.

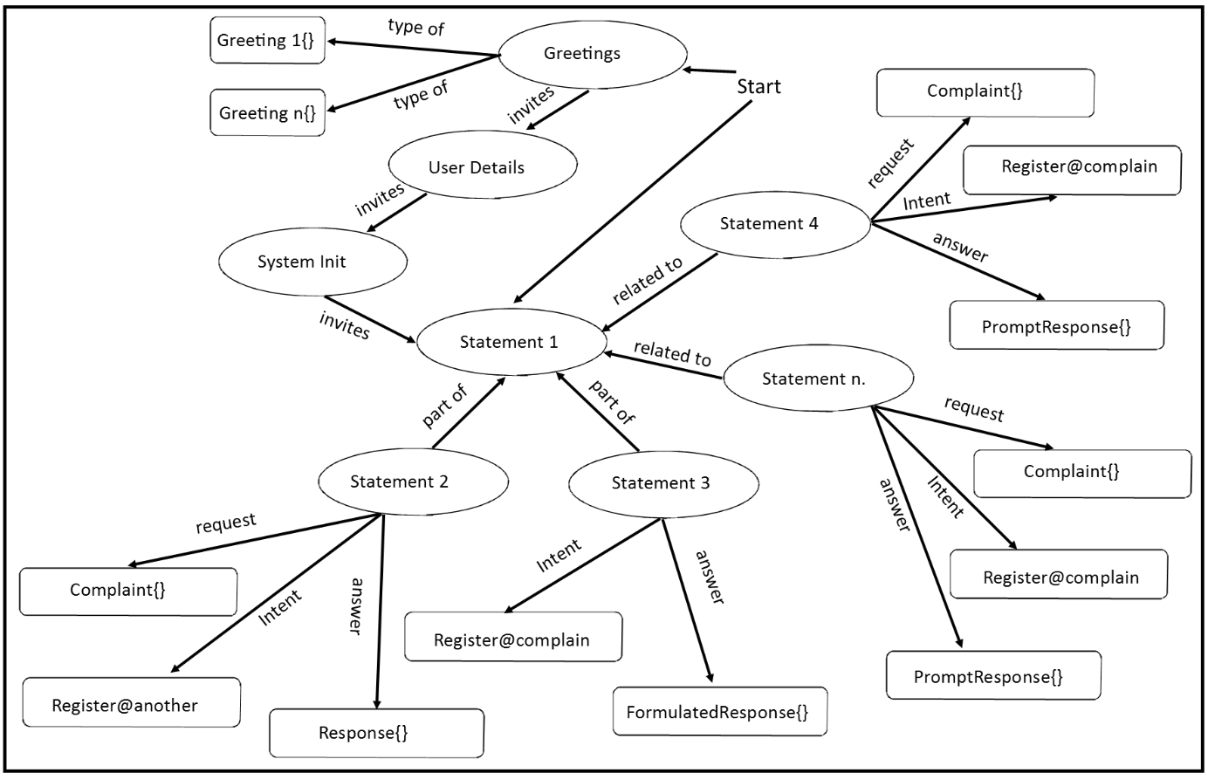


Figure 4.13: Context aware "advancing" conversation from Mshauri-Wako, in a multi-turn dialogue. In the dialogue, statements 3,4 to ...n. portray the progression of a dialogue is guaranteed and not by chance. Any scenario where the dialog follows that path, the progression is expected.

It can be explained as follows: conversation or dialogue paths may begin with some greetings or a search query, also known as a statement. A conversation depicting path one, starts with greetings and ensures that user details are first captured. After which, the system engages the user to begin a search. When a user makes a search request, the system keeps prompting for additional information. This prompting progresses until the system has acquired sufficient information to narrow the search. The system then responds with appropriate advice or response. In conversation path two, a user request takes the form of a statement. A statement may be independent or linked to another. The linkage is established through shared intent. In a statement, an agent identifies the nature of a request, registers the intent and prepare a possible response.

Statement 4 to... n, yield specific answers (prompt for responses) which are guided by the intent registered. Statement 3 yields formulated response with respect to the intent registered. The implication here is that the formulated response answers requests handled by statements

4 up to... n. This exposes a progression of statement 1, which corresponds to a progressive search.

A conversation depicting path two starts with a statement, implying that a user initiates the conversation. Statement 1 is the user request fetches intent from statement 2. Statement 2 responds with respect to the intent. New intent depicts a standalone search, in such a case, the answer will not depend on any other statement. This corresponds to the normal search.

Acceptable kappa statistic values vary on the context. However, there is consensus that in machine learning kappa statistic values above 0.40 might be considered exceptional, see appendix A8.2. The Chatbot achieved a task success rate of 0.936. This means the chatbot was excellent at achieving tasks. Based on the coherence, the chatbot achieved a core of 0.8903. This means the Chatbots ability to advance a conversation was near perfect or excellent. The overall performance of the chatbot was 0.553. This means the chatbot was good for its purpose, with respect to universally acceptable global standards.

Test objective 1.

The first testing objective was to ascertain whether “A TODS with advancing conversation ability will be judged to have a more natural conversation and has a high task success rate.” This testing objective was tested by using task success score. Figure 4.2 (c) shows the task success score of 0.936 which is interpreted as substantial or excellent. Therefore, it was concluded that “A TODS with advancing conversation ability will be judged to have a more natural conversation and has a high task success rate.”

Test objective 2.

The second test objective was to ascertain whether “A TODS with advancing conversation ability will be judged as relevant and lead the user towards the realization of a conversation goal.” This testing objective was tested, Table 4.15, metric 6, assessed the realization of the conversation goal and scored 78%. This score is considered excellent. Therefore, it is concluded that “A TODS with advancing conversation ability will be judged as relevant and lead the user towards the realization of a conversation goal.”

Test objective 3.

The third test objective was to ascertain whether “TODS with advancing conversation ability will be perceived to be more easy to use and will lead to better user satisfaction.” This testing objective was tested by metric 4 and 5 in Table 4.15. Metric 5 assessed ease and Enjoyability of interaction scored 78 %. Metric 4, as in table 4.16, assessed willingness to use system scored 80%. The scores in both cases are considered excellent. Therefore, it is concluded that "TODS with advancing conversation ability will be perceived to be more easy to use and will lead to better user satisfaction."

In reference to both qualitative and quantitative results obtained. It can be agreed that the Mshauri_Wako, the Chatbot with this embedded ability to guarantee progressive conversation, met global acceptable standards and was good for use in domain of application that demand progressive information acquisition and retrieval.

This confirms the suitability of MAS_DM architecture towards guaranteeing logically advancing conversation. Therefore, the objective of the study is satisfied.

The purpose of presenting Mitsuku was not to achieve a one on one comparison, but to guide in understanding the conversation progression. The study acknowledges that different Chatbots are built to achieve different things and thus such a direct comparison can be misleading. During the testing each participant or tester was required to try the bot at least four times before filling an online survey.

CHAPTER FIVE

CONCLUSION CONTRIBUTION AND RECOMMENDATIONS

The aim of this chapter is to present the conclusions drawn from the results of the analysis of the questionnaires, AVM scenarios, confusion matrix, system logs and computation performed. Then make recommendations for further research.

5.1 Conclusions

This thesis examined a problem in conversation that render TODS unable to serve in new domains, most of which have complex requirements. The study identified one conversational aspect that is mandatory in these new domains. The conversational aspect is advancing conversations that are logical to facilitate the achievement of a goal. The ability to support logical advancing conversations, in a way that can be guaranteed and replicated in other domains is the problem.

The study exposed that the solution relied on the dialogue management mechanisms, hence a solution required reconstruction or enhancement to dialogue management mechanisms. In that context, since the dialogue management mechanism is determined by the architecture, a promising way is to tackle the problem from the viewpoint of the underlying architecture.

The study showed that common architectures were unable to provide a solution to the problem in a way that can be learnt and easily transferred to other domains. The common ways included trying different existing architectures and using hand-crafted rules to support the adopted architecture. The study exposed why the common ways could not amicably solve the problem or even promise a solution.

The study showed that a more versatile architecture that did not depend on handcrafted rules could be easy to learn and easy to port to new domains is required. The study provided a novel architecture, MAS_DM architecture to be tried and tested. The promising results that showed that an agent-based architecture and especially a multi-agent one is suitable to guarantee a solution. The results further showed that reinforcement learning applied to multi-agent system is a promising way to guarantee advancing conversations in dialog systems. In summary, there is convincing evidence of the relevance and appropriateness of MAS_DM

architecture towards guaranteeing logically advancing conversation. The study provided proof through an artefact, a prototype dialog system for maternal healthcare diagnosis.
While the study takes cognizance of the challenges in evaluating dialog systems; - such as lack of comprehensive evaluation frameworks, lack of open-source mature or complete toolkits and subjectivity from human users, the thesis makes an important contribution to conversational artificial intelligence.

The study classifies its contribution as theoretical, methodical and contribution to practice. In theoretical contribution the study defined of a novel agent-based architectural model, (MAS_DM) for task-oriented dialog systems, along with its implementation, to provide a solution to the problem identified in the study. In methodical contribution, the study demonstrated the practicability of combining multi-agent systems and machine learning toward solving issues in conversational artificial intelligence, as demonstrated in MAS_DM architecture. In contribution to practice, the study unveiled contemporary task-oriented dialog systems equipped with new capabilities that have the potential to optimize information acquisition through assuring the logical progression of exchanges. Demonstrated through an artefact, a prototype dialog system for maternal healthcare diagnosis; - to be applied in newer domains they didn't serve before.

Work in this research provides important insights into conversational artificial intelligence. It provides an avenue for computer scientists to interrogate other conversational aspects with the objective of achieving an almost natural conversation in dialog systems. Besides, it creates the avenue of the architecture to be improved or utilized to address other conversational issues.

5.2 Recommendations for future research

The work in this thesis can be viewed as an attempt to bridge natural conversation and dialog control, it is at the frontier of conversations in dialog systems research. There is still a long way to go to create a fully fledged conversational AI with 100% natural conversation.

The research that has been undertaken for this thesis has highlighted several researchable aspects and a number of issues on which further research would be beneficial. Several issues with regards to the architecture where information is lacking were highlighted in the literature review. Whilst some of these were addressed by the research in this thesis, others remain. In particular, there are domains where dialog systems have not served before because of the complexities of the requirements. To tap into those domains there's, a need to address conversational complexities one at a time. Future studies might need to address another aspect of the conversation.

Additional issues for further research include: -The development of complete open-source toolkits that offers support to agency. The development of machine learning engines that can understand some African names. Addressing interoperability issues need to enable a seamless integration of features developed from other toolkits. The need to develop more data corpus for more domains, e.g. maternal health, telecommunications. Finally, more evaluation methods for task-oriented dialog systems need to be discussed or developed.

REFERENCES

[1] R. Banchs and H. I. Li, "A chat-oriented dialogue system based on the vector space model," in *Association for Computational Linguistics*, Jeju, 2012.

[2] H. M. Sugiyama, T. R. Higashinaka and Y. Minami, "Open-domain utterance generation for conversational dialogue systems using web-scale dependency structures," in *Proceedings of the SIGDIAL 2013 Conference* , Metz, 2013.

[3] S. Kim, C. Lee, S. Jung and G. Lee, "A spoken dialogue system for electronic program guide information access.," in *The 16th IEEE International Symposium*, 2007.

[4] J. Woo and N. Kubota, "Conversation system based on computational intelligence for robot partner using smart phone.," in *2013 IEEE International Conference*, 2013.

[5] R. Kurzweil, "The singularity is near," Gerld Duckworth & Co, 2010.

[6] R. Lopez-Cozar, Z. Callejas, D. Griol and J. F. Quesada, "Review of spoken dialogue systems," *Loquens*, vol. 1, no. 2, 2014.

[7] Forbes, "the-rise-of-conversational-ai," forbestechcouncil, 2017. [Online]. Available: https://www.forbes.com/sites/forbestechcouncil/2017. [Accessed july 2021].

[8] H. Dubberly and P. Pangaro, "what is conversation? How can we design for effective conversation?," *Interactions Magazine*, p. 9, 2009.

[9] G. M.Boyd, "Reflections on the conversation theory of Gordon Pask.," *The International Journal of Systems and Cybernetics*, vol. 30, no. 5–6, p. 560–570, 2001.

[10] G. Pask, "A guide to theory and research," in *The Cybernetics of Human Learning and Performance*, First ed., London, Hutchinson Educational, 1975, pp. 179-195.

[11] K. Mugoye, O. H. Okoyo and O. McOyowo, "Integrating Human Conversation Models Towards Improving Interaction In Text Based Dialog Systems," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, vol. 3, no. 5, pp. 539-544, 2018.

[12] H. Trieu, H. Lida, N. Bao and L. Nguyen, "Towards Developing Dialogue Systems with Entertaining Conversations," in *Proceedings of the 9th International Conference*, Nomi, 2017.

[13] G. Weisz, P. Budzianowski, P. Su and M. Gasi, "Efficient deep reinforcement learning for dialogue systems with large action spaces," *Speech and Language Processing*, vol. 1, 2018.

[14] A. Stolcke, "The SRI March 2000 Hub-5 conversational speech transcription system," in *Proceedings of NIST Speech Transcription Workshop*, 2000.

[15] S. Stoyanchev, "Exploring Adaptation in Dialog Systems," 2017.

[16] M. Sun, "Adapting Spoken Dialog Systems towards Domains and Users," 2016.

[17] M. F. McTear, Spoken Dialogue Technology, Verlag-London : Springer, 2004.

[18] E. Barnard, A. Halberstadt, C. Kotelly and M. Phillips, "A Consistent Approach To Designing Spokendialog Systems," in *Proceedings of ASRU'99 Conference*, Keystone, Colorado, 1999.

[19] Guoying Liu, "A task ontology model for domain independent dialogue management," in Proceedings of 2012 IEEE International Conference on Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2012.

[20] D. Guan, M. Chu and Q. Zhang, "The research project of man-computer dialogue system in Chinese," in *International Conference on Spoken Language Processing*, 1998.

[21] W. Shi, T. Zhao and Z. Yu, "Unsupervised dialog structure learning," in *North American Chapter of the Association for Computational Linguistics*, Minneapolis, 2019.

[22] M. McTear, "enabling the conversational user interface," *Spoken dialogue technology*, vol. 34, no. 1, p. 90–169, 2002.

[23] C. Bennett, A. Llitjos, S. Shriver, Rudnicky and A. Black, "Building Voicexml-based applications," in *Proceedings of ICSLP-2002*, Denver, Colorado, 2002.

[24] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," Cambridge Univ Press, 1998.

[25] "The Philips automatic train timetable information system," *Speech Communication*, vol. 17, no. 3-4, pp. 249-262, 1995.

[26] B. Zeigler and B. Mazor, "Dialog Design for a Speech-interactive Automation System," in *EUROSPEECH*, Madrid, Spain, 1995.

[27] C. R. Perrault, J. F. Allen and P. R. Cohen, "Speech acts as a basis for understanding dialogue coherence," in *Theoretical Issues in Natural Language Processing*, Urbana,, 1978.

[28] B. J. Grosz and C. L. Sidner, "Attention, intentions, and the structure of discourse," *Computational linguistics*, vol. 12, no. 3, pp. 175-204, 1986.

[29] G. Skantze, "Error Handling in Spoken Dialogue Systems-Managing Uncertainty, Grounding and Miscommunication," Stockholm, 2007.

[30] X. Wu, F. Zheng and M. Xu, "Topic forest: A plan-based dialog management structure.," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, 2001.

[31] J. Chu-Carroll and S. Carberry, "A plan-based model for response generation in collaborative task-oriented dialogues," in *12th National Conference on Artificial Intelligence*, Seattle, 1994.

[32] J. D. Moore and C. L. Paris, "Planning text for advisory dialogues: Capturing intentional and rhetorical information," *Computational linguistics*, vol. 19, no. 4, pp. 651-694, 1993.

[33] K. Wang, "A plan-based dialog system with probabilistic inferences," in *INTERSPEECH Annual Conference*, Beijing.

[34] L. L, D. AG, T. HL, K. AB, C. J, B. R, S. D, G. B, M. F and L. AY, "Conversational agents in healthcare: a systematic review," *J Am Med Inform Assoc*, vol. 25, no. 9, p. 1248–58, 2018.

[35] A. Nguyen and W. Wobcke, "An Agent-Based Approach to Dialogue Management inPersonal Assistants," in *International Conference on Intelligent User Interfaces*, San Diego, California, 2005.

[36] N. M and H. A, "Agent-based computing from multi-agent systems to agent-based models. A visual survey," *Scientometrics*, vol. 89, no. 2, pp. 479-499, 2011.

[37] W. M, in *An Introduction to Multi-Agent Systems*, England, John Wiley & Sons, 2009, p. 453.

[38] N. Blaylock and G. Ferguson, "Synchronization in an Asynchronous Agent-Based Architecture for Dialogue Systems.," in *3rd SIGdial Workshop on Discourse and Dialog*, 2002.

[39] D. Litman and S. Pan, "Designing and evaluating an adaptive spoken dialogue system.," *User Modeling and User-Adapted Interaction*, vol. 12, no. 2-3, p. 111–137, 2002.

[40] M. Ghazvininejad, C. Brockett, M. Chang, B. Dolan, J. Gao, W. Yih and M. Galley, "A Knowledge-Grounded Neural Conversation Model," in *Proceedings of 32nd AAAI Conference*, 2018.

[41] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao and D. Jurafsky, "Deep Reinforcement Learning for Dialogue Generation," in *Empirical Methods in Natural Language Processing*, Austin, 2016.

[42] S. Singh, D. Litman, M. Kearns and M. Walker, "Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system.," *Journal of Artificial Intelligence Research*, vol. 16, p. 105–133, 2002.

[43] M. Walker, "Dynamic Adaptation in Dialog Systems," in Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Tokyo, 2010.

[44] J. Baskar, "Adaptive Human-Agent Dialogues for Reasoning about Health," 2014.

[45] M. Gnjatovi´c and D. R¨osner, "Adaptive dialogue management in the nimitek prototype system.," in Proceedings of the 4th IEEE tutorial and research workshop on Perception and Interactive Technologies for Speech-Based Systems, Berlin, 2008.

[46] A. Jameson, "Adaptive Interfaces and Agents.," in *Human-Computer Interface Handbook*, 2003, pp. 305-330.

[47] P. Johansson, "Design and Development of Recommender Dialogue Systems," *Licentiate Thesis. Linköping Studies in Science and Technology,* 2004.

[48] F. Nothdurft, F. Honold and P. Kurzok, "Using explanations for runtime dialogue adaptation," in *Proceedings of the 14th ACM international conference on Multimodal interaction*, 2012.

[49] S. Worswick, july 2020. [Online]. Available: https://web.facebook.com/mitsukubot/?_rdc=1&_rdr.

[50] V. Rieser and O. Lemon, "Reinforcement Learning for Adaptive Dialogue Systems," 2011.

[51] P. A. Inc, "Potentia Analytics," [Online]. Available: https://www.potentiaco.com/what-is-machine. [Accessed jan 2021].

[52] D. Litman, M. Kearns and M. Walker, "Optimizing dialogue management with reinforcement learning," *Journal of Artificial Intelligence Research,* vol. 16, pp. 105-133, 2002.

[53] R. V. Solingen, V. Basili, G. Caldiera and H. D. Rombach, "Goal Question Metric (GQM) Approach.," in *Encyclopedia of Software Engineering*, 2002.

[54] M. A. Walker, D. J. Litman, C. A. Kamm and A. Abella, "PARADISE: A Framework for Evaluating Spoken Dialogue Agents," in *ACL/EACL 35th Annual Meeting of the Association for Computational Linguistics*, San Francisco, 1997.

[55] A. Abran, A. Khelifi and W. Suryn, "Usability meanings and interpretations in ISO," *Software Quality Journal*, vol. 11, p. 325–338, 2003.

[56] J. C. Carletta, "Assessing the reliability of subjective codings," in *Computational Linguistics*, 1996.

[57] S. Siegel and N. J. Castellan, Nonparametric Statistics for the Behavioral Sciences, 2 ed., New York: McGraw-Hill, 1988.

[58] H. Cuayahuitl, S. Renals, O. Lemon and H. Shimodaira, "Evaluation of a hierarchical reinforcement learning," *Science Direct Computer Speech and Language*, vol. 24, p. 395–429, 2010.

[59] C. Kothari, Research Methodology: Methods and Techniques, New Delhi: New Age International Publishers, 2004.

[60] J. N. Amaral, "José Nelson Amaral's Publications," 28 November 2017. [Online]. Available:

http://webdocs.cs.ualberta.ca/~c603/readings/researchmethods.pdf. [Accessed 28 November 2017].

[61] M. M. Ayash, "Research Methodologies in Computer Science and Information Systems," 28 November 2017. [Online]. Available: http://www.ptcdb.edu.ps/ar/sites/default/files. [Accessed 28 November 2017].

[62] S. J. Russell and P. Norvig, "A Modern Approach Third Edition," in *Artificial Intelligence*, New Jersey, Pearson Education, Inc., 2003, pp. 54-59.

[63] M. Berndtsson, J. Hansson, B. Olsson and B. Lundell, Thesis Projects , A Guide for Students in Computer Science and Information Systems, springer , 2008 .

[64] F. Davis and V. Venkatesh, "Toward pre-prototype user acceptance testing of new information systems: implications for software project management'," *IEEE Transactions on Engineering Management*, vol. 5, no. 1, pp. 31- 46, 2004.

[65] K. Mugoye, H. Okoyo and S. McOywo, "Smart-bot Technology: Conversational Agents Role in Maternal Healthcare Support," in *IST-Africa 2019 Conference Nairobi*, Nairobi, 2019.

[66] K. Mugoye, O. Okoyo and O. McOyowo, "MAS architectural model for dialog systems with advancing conversations," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 3, no. 8, pp. 247-252, 2018.

[67] Google, "Build natural and rich conversational experiences," Google LLC, [Online]. Available: https://dialogflow.com/. [Accessed 29 April 2019].

[68] m. kevin, "demo-bot-world," 2018. [Online]. Available: https://sites.google.com/view/demo-bot-world/home?read_current=1.

[69] W. Chin and P. Newsted, "Structural Equation Modeling analysis with Small Samples using Partial Least Squares," in *Statistical Strategies for Small Sample Research*, California, 1999.

[70] W. G. Cochran, Sampling techniques, New York: John Wiley & Sons, 1977.

[71] E. Overby and B. Konsynski, "An integrated model and empirical test," *Task-Technology fit and process virtualization theory*, 2010.

[72] D. Goodhue, W. Lewis and R. Thompson, "small sample size, and statistical power in MIS research," in *International Conference on*, Hawaii, 2006.

[73] W. L. Chapman, W. J. Welch, K. P. S. J. Bowman and J. E. Walsh, "Arctic Sea Ice Variability: Model Sensitivities and a Multidecadal Simulation," *Journal of Geophysical Research*, vol. 99, p. 919–935, 1994.

[74] D. Jones, M. Schonlau and W. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, vol. 13, p. 455–492, 1998.

[75] S. T. R. and D. N. T., "Sample size considerations and augmentation of computer experiments," *Journal of Statistical Computation and Simulation,* vol. 68, no. 4, pp. 307-319, 2001 .

[76] J. Cohen, "The earth is round (p < .05): Rejoinder," *American Psychologist,* vol. 50, p. 1103, 1995.

[77] J. Landis and G. Koch, "The measurement of observer agreement for categorical data," *Biometrics,* vol. 33, no. 1, p. 159–174, 1977.

[78] J. Fleiss, Statistical methods for rates and proportions, second ed., New York: John Wiley, 1981.

[79] R. Steinbiss, B. Haeb-Umbach and U. E. Tran, "The Philips automatic train timetable information system," *Speech Communication,* vol. 17, no. 3-4, pp. 249-262, 1995.

[80] "Build natural and rich conversational experiences," Google, [Online]. Available: https://dialogflow.com/ . [Accessed 29 April 2019].

[81] G. D, M. H, P. J, S. S and B. S, "A form-based dialogue manager for spoken language applications," in *ICSLP*, 1996.

[82] Z. V, S. S, G. J, P. J, P. C, H. T.J and H. L, " A telephone-based conversational interface for weather information," *Jupiter*, vol. 8, pp. 85-96, 2000.

[83] S. S. and P. J., "Dialogue management in the mercury flight reservation," in *ANLP-NAACL Workshop on Conversational Systems*, 2000.

[84] ravenclaw, "Olympus," [Online]. Available: www.ravenclaw-olympus.org.

[85] charlesrich, "Disco," [Online]. Available: https://github.com/charlesrich/Disco.

[86] R. Pieraccini, D. Suendermann, K. Dayanidhi and J. Liscombe, "Are We There Yet? Research in Commercial Spoken Dialog Systems," in *Text, Speech and Dialogue*, 2009.

[87] L. D., Z. S., S. P. and C. X., "Leveraging commonsense reasoning and multimodal perception for robot spoken dialog systems," in *International Conference on Intelligent Robots and Systems (IROS)*, 2017.

APPENDICES APPENDIX 1: SUMMARY OF USER SURVEY DATA

What is your Age? (in years)		
Answer Choices	Responses	
19-30	56.67%	34
31-49	30.00%	18
50 or more	13.33%	8
	Answered	60
	Skipped	0

Table A (1)

Specify your Gender		
Answer Choices	Responses	
Female	68.33%	41
Male	31.67%	19
	Answered	60
	Skipped	0

Table B (1)

The system was efficient in accomplishing the task		
Answer Choices	Responses	
Never	5.00%	3
Rarely	10.00%	6
Sometimes	21.67%	13
Often	33.33%	20
Always	30.00%	18
	Answered	60
	Skipped	0

Table C (1)

The system can detect intent or meaning.		
Answer Choices	Responses	
Never	6.67%	4
Rarely	6.67%	4
Sometimes	33.33%	20
Often	20.00%	12
Always	33.33%	20
	Answered	60
	Skipped	0

Table D (1)

The system conveys personality.		
Answer Choices	Responses	
Never	6.67%	4
Rarely	13.33%	8
Sometimes	20.00%	12
Often	16.67%	10
Always	43.33%	26
	Answered	60
	Skipped	0

Table E (1)

The system makes the conversation easier.		
Answer Choices	Responses	
Never	3.33%	2
Rarely	11.67%	7
Sometimes	11.67%	7
Often	31.67%	19
Always	41.67%	25
	Answered	60
	Skipped	0

Table F (1)

The system is easy to use.		
Answer Choices	Responses	
Never	1.67%	1
Rarely	6.67%	4
Sometimes	16.67%	10
Often	21.67%	13
Always	53.33%	32
	Answered	60
	Skipped	0

Table G (1)

The system maintains a natural satisfying interaction.		
Answer Choices	Responses	
Never	6.67%	4
Rarely	5.00%	3
Sometimes	16.67%	10
Often	21.67%	13
Always	50.00%	30
	Answered	60
	Skipped	0

Table H (1)

The system solicits information logically.		
Answer Choices	Responses	
Never	1.67%	1
Rarely	5.00%	3
Sometimes	21.67%	13
Often	23.33%	14
Always	48.33%	29
	Answered	60
	Skipped	0

Table I (1)

The system maintains the theme of discussion.		
Answer Choices	Responses	
Never	1.67%	1
Rarely	0.00%	0
Sometimes	13.33%	8
Often	21.67%	13
Always	63.33%	38
	Answered	60
	Skipped	0

Table J (1)

The system quickly provided the information that I needed.		
Answer Choices	Responses	
Never	5.00%	3
Rarely	3.33%	2
Sometimes	13.33%	8
Often	23.33%	14
Always	55.00%	33
	Answered	60
	Skipped	0

Table K (1)

The system has a fluent dialogue.		
Answer Choices	Responses	
Never	0.00%	0
Rarely	6.67%	4
Sometimes	16.67%	10
Often	26.67%	16
Always	50.00%	30
	Answered	60
	Skipped	0

Table L (1)

I enjoyed the time that I spent using the system.		
Answer Choices	Responses	
Never	0.00%	0
Rarely	6.67%	4
Sometimes	16.67%	10
Often	20.00%	12
Always	56.67%	34
	Answered	60
	Skipped	0

Table M (1)

I would love to use the system again.		
Answer Choices	Responses	
Never	0.00%	0
Rarely	3.33%	2
Sometimes	16.67%	10
Often	13.33%	8
Always	66.67%	40
	Answered	60
	Skipped	0

Table N (1)

APPENDIX 2: SUMMARY OF CODED DATA FROM USER SURVEY

user	US	k	AC	AF	FX	H	NUU
1	5	1	5	5	5	5	28
2	5	1	5	5	5	5	36
3	5	1	5	5	5	5	32
4	5	1	5	5	5	5	30
5	5	1	5	5	5	5	31
6	5	1	5	5	5	5	32
7	5	1	5	5	5	5	28
8	5	1	5	5	5	5	28
9	5	1	5	5	5	5	36
10	5	1	5	5	5	5	38
11	5	1	5	5	5	5	32
12	5	1	5	5	5	5	34
13	5	1	5	5	5	5	30
14	5	1	5	5	5	5	34
15	5	1	5	5	5	5	32
16	5	1	5	5	5	5	34
17	5	1	5	5	5	5	30
18	5	1	5	5	5	5	32
19	5	1	5	5	4	5	34
20	5	1	5	5	4	5	34
21	5	1	5	5	4	5	38
22	5	1	5	5	4	5	36
23	5	1	5	5	4	5	28
24	5	1	5	5	4	5	28
25	5	1	5	5	4	5	30
26	5	1	5	5	4	5	32
27	5	1	5	4	4	5	24
28	5	1	5	4	4	5	23
29	5	1	5	4	4	5	22
30	5	1	5	4	4	5	26
31	5	1	5	4	4	5	26
32	5	1	5	4	4	5	20
33	5	1	5	4	4	5	20
34	5	1	5	4	4	5	18
35	5	1	5	4	4	5	18
36	5	1	5	4	4	5	18
37	5	1	5	3	4	5	16

38	5	1	5	3	4	5	14
39	5	1	5	3	4	4	16
40	5	1	5	3	4	4	12
41	4	1	5	3	4	4	14
42	4	1	4	3	4	4	14
43	4	1	4	3	4	4	12
44	4	1	4	3	4	4	14
45	4	1	4	3	3	4	16
46	4	1	4	3	3	4	14
47	4	1	4	3	3	3	12
48	4	1	4	3	3	3	14
49	3	0.936	4	2	3	3	12
50	3	0.936	4	2	3	3	14
51	3	0.936	4	2	3	3	14
52	3	0.936	4	2	2	3	14
53	3	0.936	4	2	2	3	12
54	3	0.936	4	2	2	3	14
55	3	0.936	4	2	2	3	12
56	3	0.936	4	2	2	3	12
57	3	0.936	4	2	2	3	12
58	2	0.936	1	1	2	3	6
59	2	0.936	1	1	3	2	6
60	2	0.936	1	1	1	1	6
Mean	4.416667	0.9872	4.533333	3.783333	3.883333	4.35	22.567
SDEV		0.025816	0.929127	1.290009	1.026623	0.971195	9.501769

Table A2. Summary of coded data from user survey

APPENDIX 3: NORMALIZED ATTRIBUTE SCORES AND PERFORMANCE RATING

AF	FX	H	NUU	AF	FX	H	NUU
5	5	5	28	0.93465382	0.791819	0.645377	0.571823
5	5	5	36	0.93465382	0.791819	0.645377	1.413772
5	5	5	32	0.93465382	0.791819	0.645377	0.992798
5	5	5	30	0.93465382	0.791819	0.645377	0.78231
5	5	5	31	0.93465382	0.791819	0.645377	0.887554
5	5	5	32	0.93465382	0.791819	0.645377	0.992798
5	5	5	28	0.93465382	0.791819	0.645377	0.571823
5	5	5	28	0.93465382	0.791819	0.645377	0.571823
5	5	5	36	0.93465382	0.791819	0.645377	1.413772
5	5	5	38	0.93465382	0.791819	0.645377	1.624259
5	5	5	32	0.93465382	0.791819	0.645377	0.992798
5	5	5	34	0.93465382	0.791819	0.645377	1.203285
5	5	5	30	0.93465382	0.791819	0.645377	0.78231
5	5	5	34	0.93465382	0.791819	0.645377	1.203285
5	5	5	32	0.93465382	0.791819	0.645377	0.992798
5	5	5	34	0.93465382	0.791819	0.645377	1.203285
5	5	5	30	0.93465382	0.791819	0.645377	0.78231
5	5	5	32	0.93465382	0.791819	0.645377	0.992798
5	5	5	34	0.93465382	0.791819	0.645377	1.203285
5	5	5	34	0.93465382	0.791819	0.645377	1.203285
5	5	5	38	0.93465382	0.791819	0.645377	1.624259
5	5	5	36	0.93465382	0.791819	0.645377	1.413772
5	5	5	28	0.93465382	0.791819	0.645377	0.571823
5	5	5	28	0.93465382	0.791819	0.645377	0.571823
5	5	5	30	0.93465382	0.791819	0.645377	0.78231
5	5	5	32	0.93465382	0.791819	0.645377	0.992798
4	5	5	24	0.176826398	0.791819	0.645377	0.150849
4	5	5	23	0.176826398	0.791819	0.645377	0.045606
4	5	5	22	0.176826398	0.791819	0.645377	-0.05964
4	5	5	26	0.176826398	0.791819	0.645377	0.361336
4	4	5	26	0.176826398	-0.0273	0.645377	0.361336
4	4	5	20	0.176826398	-0.0273	0.645377	-0.27013
4	4	5	20	0.176826398	-0.0273	0.645377	-0.27013
4	4	5	18	0.176826398	-0.0273	0.645377	-0.48061
4	4	5	18	0.176826398	-0.0273	0.645377	-0.48061
4	4	5	18	0.176826398	-0.0273	0.645377	-0.48061

3	4	5	16	-0.581001023	-0.0273	0.645377	-0.6911
3	4	5	14	-0.581001023	-0.0273	0.645377	-0.90159
3	4	4	16	-0.581001023	-0.0273	-0.52804	-0.6911
3	4	4	12	-0.581001023	-0.0273	-0.52804	-1.11207
3	4	4	14	-0.581001023	-0.0273	-0.52804	-0.90159
3	4	4	14	-0.581001023	-0.0273	-0.52804	-0.90159
3	4	4	12	-0.581001023	-0.0273	-0.52804	-1.11207
3	3	4	14	-0.581001023	-0.84643	-0.52804	-0.90159
3	3	4	16	-0.581001023	-0.84643	-0.52804	-0.6911
3	3	4	14	-0.581001023	-0.84643	-0.52804	-0.90159
3	3	4	12	-0.581001023	-0.84643	-0.52804	-1.11207
3	3	4	14	-0.581001023	-0.84643	-0.52804	-0.90159
2	3	4	12	-1.338828445	-0.84643	-0.52804	-1.11207
2	3	4	14	-1.338828445	-0.84643	-0.52804	-0.90159
2	3	4	14	-1.338828445	-0.84643	-0.52804	-0.90159
2	3	3	14	-1.338828445	-0.84643	-1.70145	-0.90159
2	3	3	12	-1.338828445	-0.84643	-1.70145	-1.11207
2	2	3	14	-1.338828445	-1.66555	-1.70145	-0.90159
2	2	3	12	-1.338828445	-1.66555	-1.70145	-1.11207
2	2	3	12	-1.338828445	-1.66555	-1.70145	-1.11207
1	1	3	12	-2.096655867	-2.48467	-1.70145	-1.11207

1	1	3	6	-2.096655867	-2.48467	-1.70145	-1.74353
1	1	3	6	-2.096655867	-2.48467	-1.70145	-1.74353
1	1	1	6	-2.096655867	-2.48467	-4.04827	-1.74353
3.766667	4.033333	4.45	22.567		0	0	0

Table A3(a) Summary of normalized data from user survey

APPENDIX 4: REGRESSION OUTPUT

SUMMARY OUTPUT								
Regression Statistics								
Multiple R	0.940151756							
R Square	0.883885324							
Adjusted R Square	0.877664895							
Standard Error	0.31128839							
Observations	60							
ANOVA								
	df	*SS*	*MS*	*F*	*Significance F*			
Regression	3	41.30690748	13.76896916	142.093948	3.73938E-26			
Residual	56	5.426425853	0.096900462					
Total	59	46.73333333						
	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	0.689791309	0.266606462	2.587301541	0.012296351	0.155714389	1.223868229	0.155714389	1.223868229
AF	0.28147159	0.086972683	3.236321812	0.002035315	0.107244371	0.45569881	0.107244371	0.45569881
FX	-0.058345803	0.111901417	-0.521403607	0.604141669	-0.282511278	0.165819673	-0.282511278	0.165819673
H	0.651945325	0.114614295	5.688167676	4.86039E-07	0.422345304	0.881545347	0.422345304	0.881545347

Figure A4.1. Initial regression output.

Second regression after removal of not-significant factor

SUMMARY OUTPUT								
Regression Statistics								
Multiple R	0.939851917							
R Square	0.883321625							
Adjusted R Square	0.879227647							
Standard Error	0.309293745							
Observations	60							
ANOVA								
	df	*SS*	*MS*	*F*	*Significance F*			
Regression	2	41.28056395	20.64028198	215.7612015	2.56582E-27			
Residual	57	5.452769379	0.095662621					
Total	59	46.73333333						
	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	0.718443699	0.25921	2.771666596	0.007519615	0.199384627	1.237502771	0.199384627	1.237502771
AF	0.250955245	0.063921296	3.926003686	0.000235317	0.122955057	0.378955433	0.122955057	0.378955433
H	0.622387613	0.098975077	6.288326613	4.82791E-08	0.42419344	0.820581787	0.42419344	0.820581787

Table B4. Regression output after removal of not-significant factor.

APPENDIX 5: EVALUATION BASED ON GQM

Goal 1	***Purpose***	***Implement a DS that support***	***Response***
	Issue	***Logically progressing***	
	Object	***Conversation***	
	Viewpoint	***From the user's viewpoint***	
Question	***Q1***	***Is the DS advancing a conversation?***	
Metrics	M1	-Support of Sub-dialog to feed into main dialog	Yes
	M2	-Occurrence of progressive exchange	Yes
	M3	-Number of correct responses	93
Question	***Q2***	***Are user satisfied?***	
Metrics	M4	-% Ease of interaction	75
	M5	-% Enjoyability of interaction	80
Question	***Q3***	***Is the architecture suitable for advancing conversation?***	
Metrics	M6	-Realization of conversation goal	78
	M7	-Naturalness of conversation	72

Table A5.1. Results after analysis of Goal 1 w.r.t GQM

Goal 2	***Purpose***	***Verify if the***	***Response***
	Issue	***DS informatively handles the***	
	Object	***conversation from***	
	Viewpoint	***the user's viewpoint***	
Question	***Q1***	***Is the exchange relevant to a user query?***	
Metrics	M1	Classification of the exchanges	yes
	M2	User perception of the conversation	Enjoyable
	M3	Number of correct responses	93
Question	***Q2***	***Does the exchange elicit more information about the query?***	
Metrics	M4	User willingness to use system again	80

Table A5.2. Results after analysis of Goal 2 w.r.t GQM

APPENDIX 6: EVALUATION-TEST QUESTIONNAIRE

See page below.

Instructions

Answer questions as they relate to you. For most answers, check the box(es) most applicable to you or fill in the blanks.

User's opinion on conversation with machines and what they would desire

Personal Details (This is for analysis only)

1. Your Age (years)

(Select only one.)

- ☐ 19-30
- ☐ 31-49
- ☐ 50 or more

2. Your Gender

(Select only one.)

- ☐ Female
- ☐ Male

Personal Opinion

Please state your level of agreement regarding the following statements by marking one of the boxes. 1 = never, 2 = rarely, 3 = sometimes, 4 = often, 5 = always.

	1	2	3	4	5
	Never....			Always	
3. Accessibility and Affect					
(Select only one box.)					
The system was efficient in accomplishing the task.	☐	☐	☐	☐	☐
The system can detect intent or meaning.	☐	☐	☐	☐	☐
The system conveys personality.	☐	☐	☐	☐	☐
The system makes the conversation easier.	☐	☐	☐	☐	☐
Interaction Experience					
4. Functionality and Humanity					
The system is easy to use.	☐	☐	☐	☐	☐
The system maintains a natural satisfying interaction.	☐	☐	☐	☐	☐
The system solicits information logically.	☐	☐	☐	☐	☐
The system maintains the theme of discussion.	☐	☐	☐	☐	☐
The system quickly provided the information that I needed.	☐	☐	☐	☐	☐
5. Meeting User Expectation					
The system has a fluent dialogue.	☐	☐	☐	☐	☐
I enjoyed the time that I spent using the system.	☐	☐	☐	☐	☐
I would love to use the system again.	☐	☐	☐	☐	☐

APPENDIX 7A: DEPLOYMENT ENVIRONMENTS

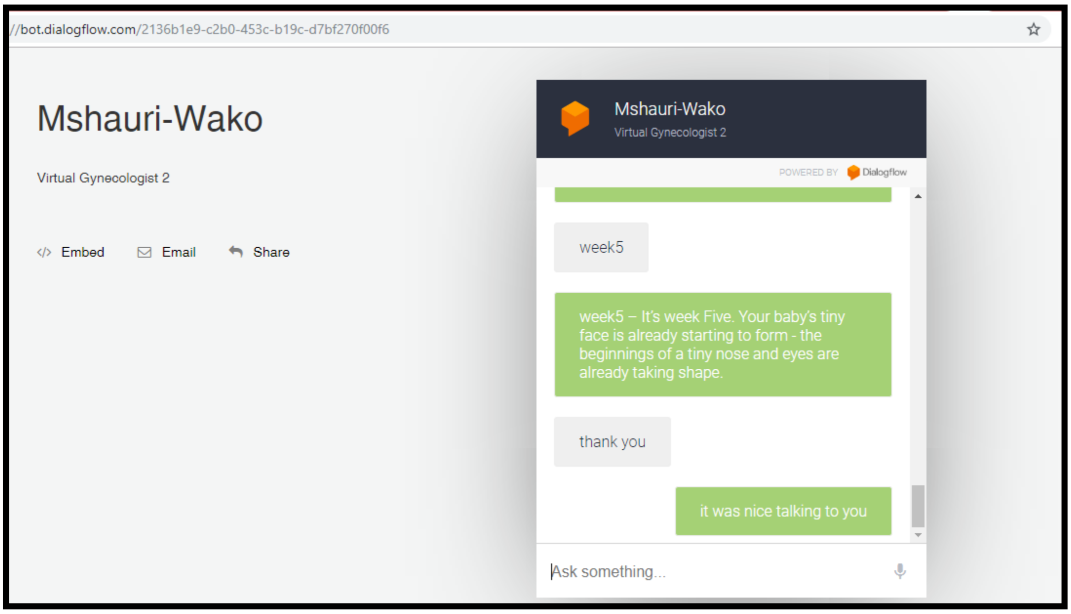


Figure A7.1. Simple web deployment of the chatbot.

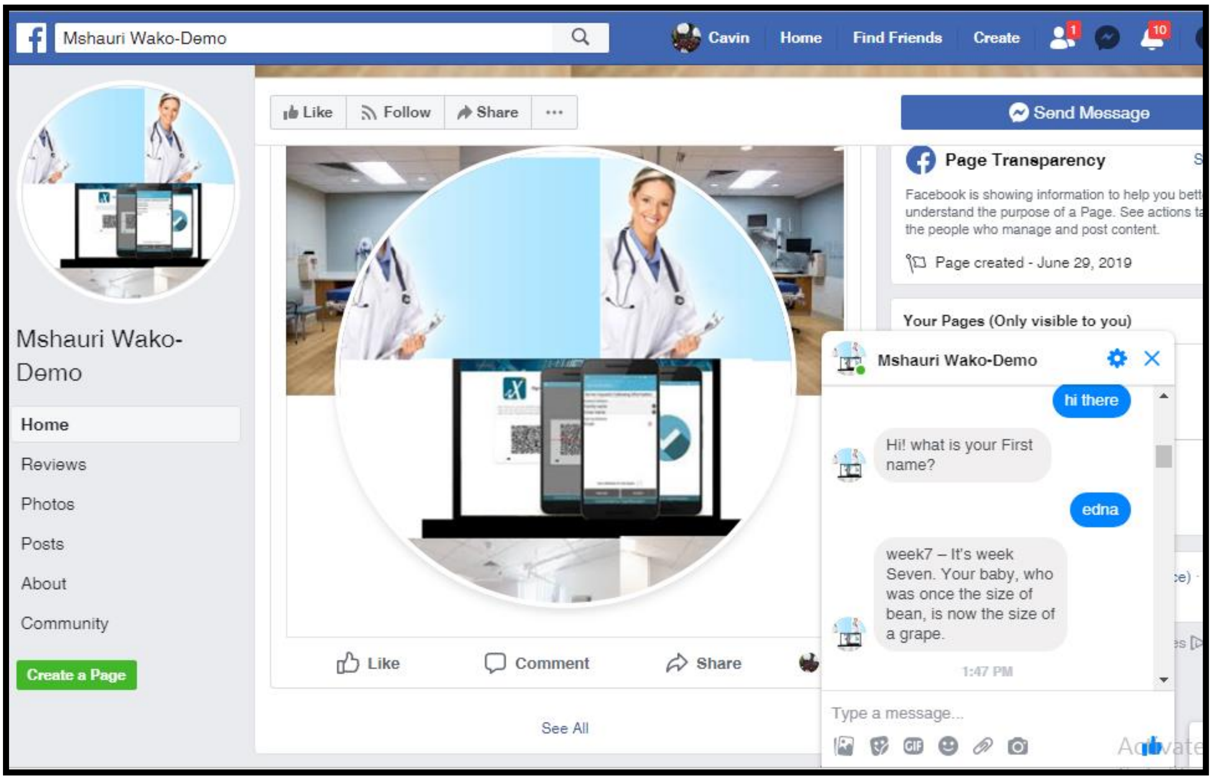
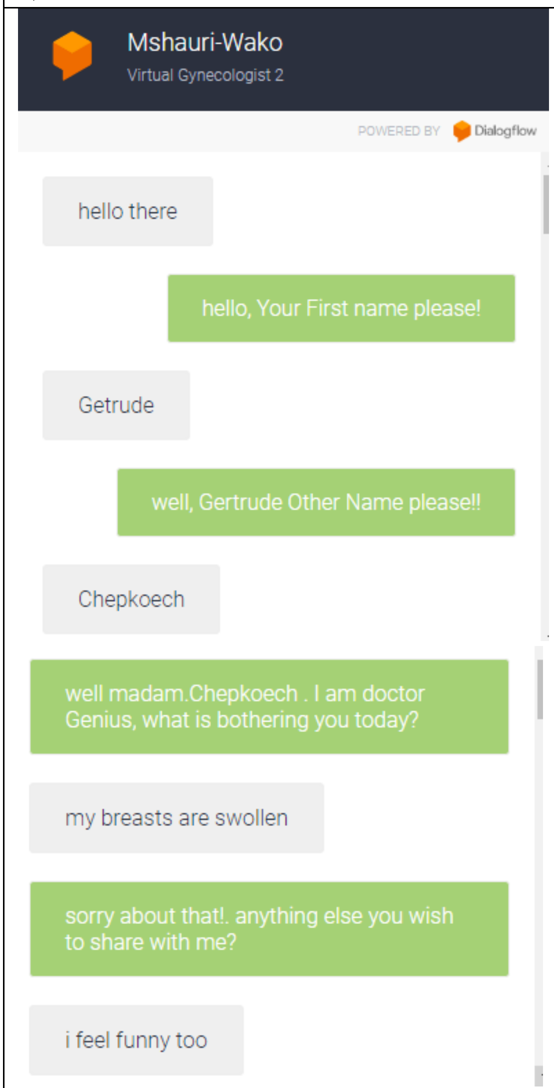
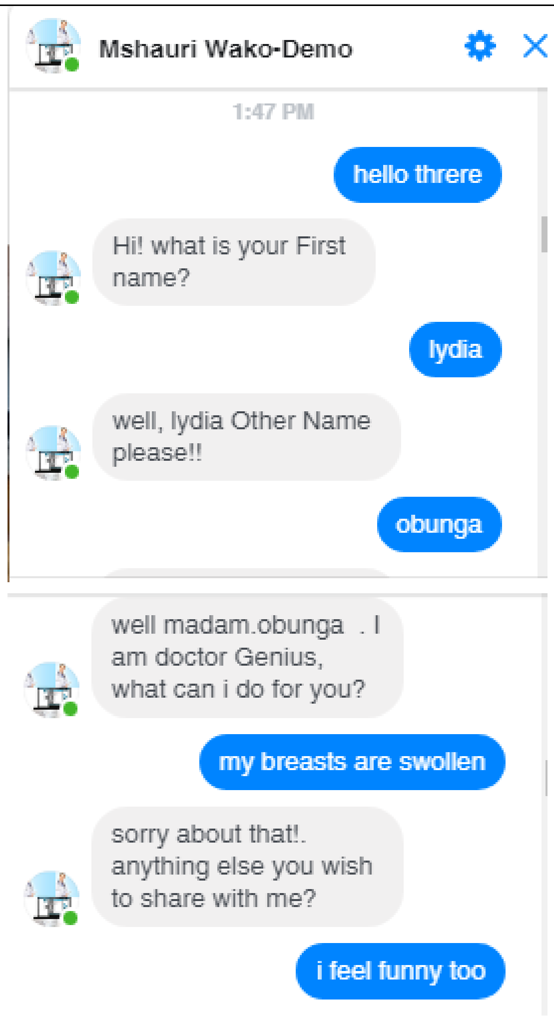
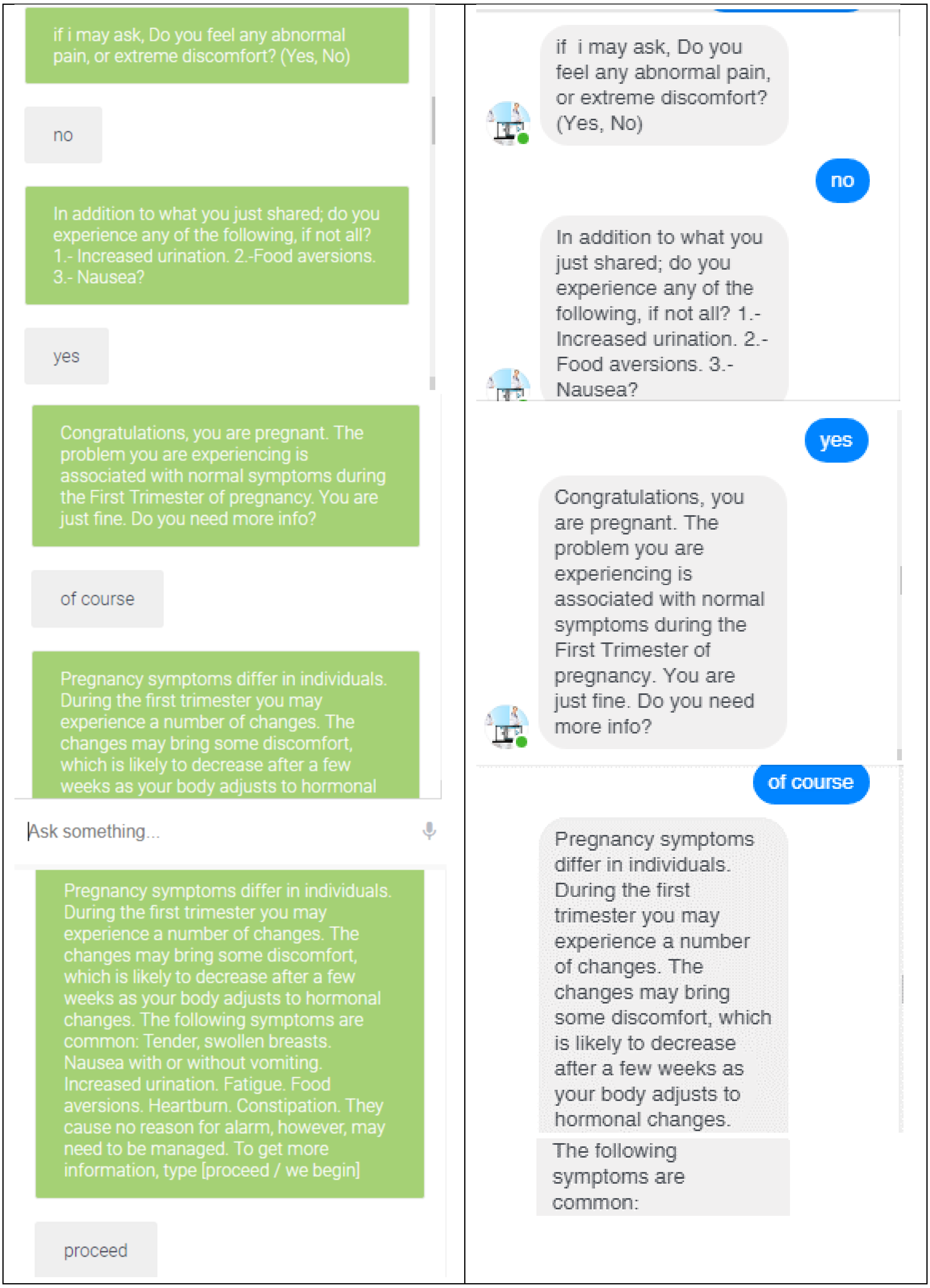


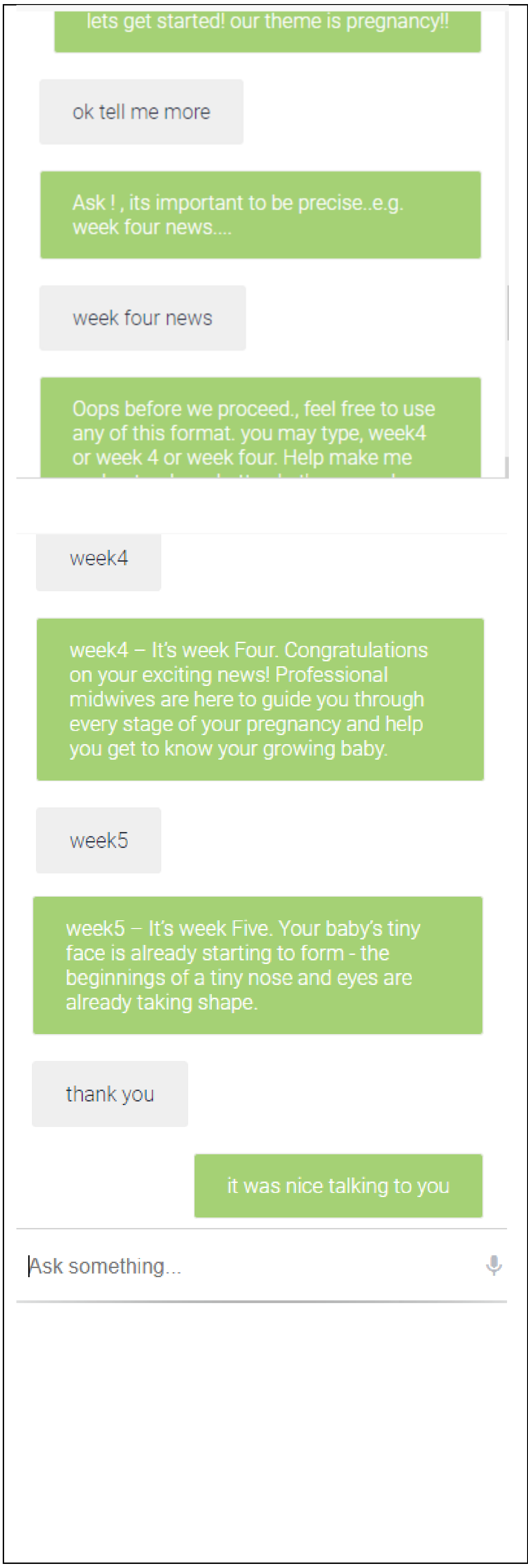
Figure A7.2. Facebook integration of the chatbot.

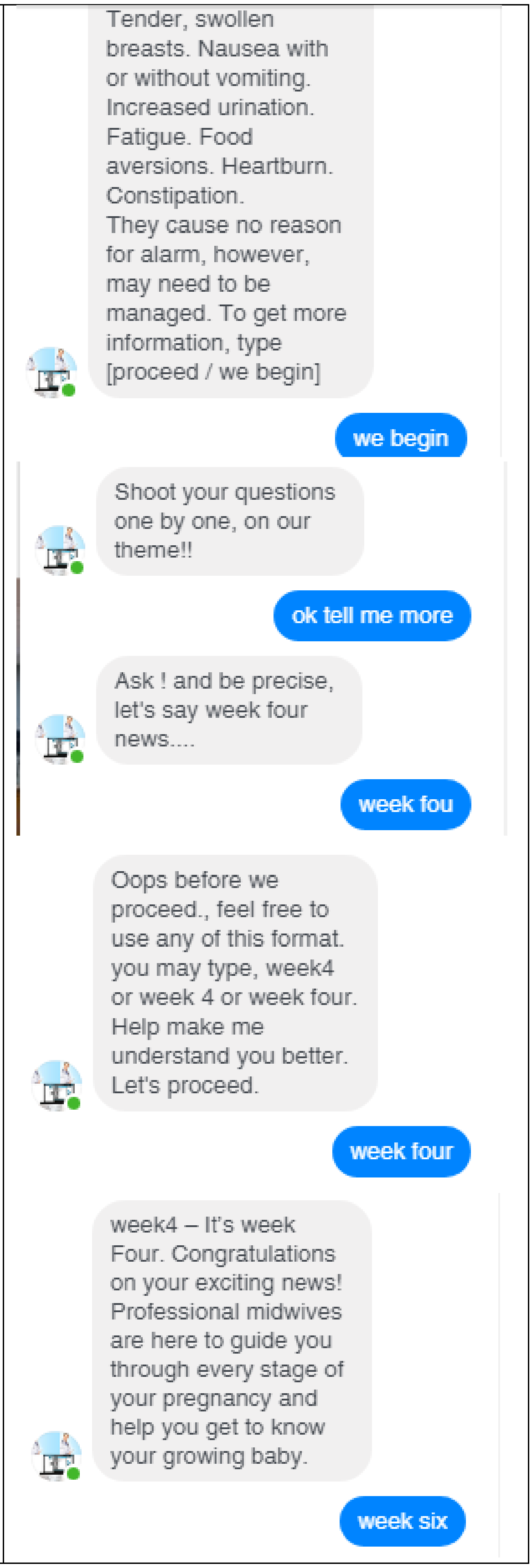
APPENDIX 7B : SAMPLE CONVERSATION FROM DIFFERENT DEPLOYMENT ENVIRONMENTS

The figure B7.1 shows a sample end-to-end conversations from Mshauri-wako chatbot within the maternal domain context. The figure further illustrates the conversation across different deployment environments. The perspective represents a reflection of user biases.

Simple web implementation	Facebook implementation
a)	b)
	


if i may ask, Do you feel any abnormal pain, or extreme discomfort? (Yes, No)
no
In addition to what you just shared; do you experience any of the following, if not all? 1.- Increased urination. 2.-Food aversions. 3.- Nausea?
yes
Congratulations, you are pregnant. The problem you are experiencing is associated with normal symptoms during the First Trimester of pregnancy. You are just fine. Do you need more info?
of course
Pregnancy symptoms differ in individuals. During the first trimester you may experience a number of changes. The changes may bring some discomfort, which is likely to decrease after a few weeks as your body adjusts to hormonal
Ask something...
Pregnancy symptoms differ in individuals. During the first trimester you may experience a number of changes. The changes may bring some discomfort, which is likely to decrease after a few weeks as your body adjusts to hormonal changes. The following symptoms are common: Tender, swollen breasts. Nausea with or without vomiting. Increased urination. Fatigue. Food aversions. Heartburn. Constipation. They cause no reason for alarm, however, may need to be managed. To get more information, type [proceed / we begin]
proceed
if i may ask, Do you feel any abnormal pain, or extreme discomfort? (Yes, No)
no
In addition to what you just shared; do you experience any of the following, if not all? 1.- Increased urination. 2.- Food aversions. 3.- Nausea?
yes
Congratulations, you are pregnant. The problem you are experiencing is associated with normal symptoms during the First Trimester of pregnancy. You are just fine. Do you need more info?
of course
Pregnancy symptoms differ in individuals. During the first trimester you may experience a number of changes. The changes may bring some discomfort, which is likely to decrease after a few weeks as your body adjusts to hormonal changes.
The following symptoms are common:


lets get started! our theme is pregnancy!!
ok tell me more
Ask ! , its important to be precise..e.g. week four news....
week four news
Oops before we proceed., feel free to use any of this format. you may type, week4 or week 4 or week four. Help make me
week4
week4 – It's week Four. Congratulations on your exciting news! Professional midwives are here to guide you through every stage of your pregnancy and help you get to know your growing baby.
week5
week5 – It's week Five. Your baby's tiny face is already starting to form - the beginnings of a tiny nose and eyes are already taking shape.
thank you
it was nice talking to you
Ask something...


Tender, swollen breasts. Nausea with or without vomiting. Increased urination. Fatigue. Food aversions. Heartburn. Constipation.
They cause no reason for alarm, however, may need to be managed. To get more information, type [proceed / we begin]
we begin
Shoot your questions one by one, on our theme!!
ok tell me more
Ask ! and be precise, let's say week four news....
week fou
Oops before we proceed., feel free to use any of this format. you may type, week4 or week 4 or week four. Help make me understand you better. Let's proceed.
week four
week4 – It's week Four. Congratulations on your exciting news! Professional midwives are here to guide you through every stage of your pregnancy and help you get to know your growing baby.
week six

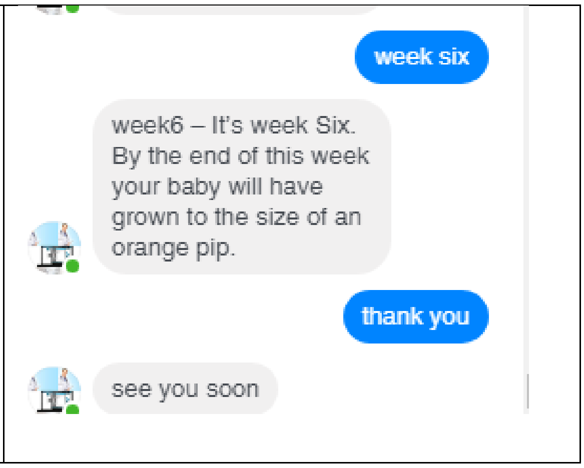
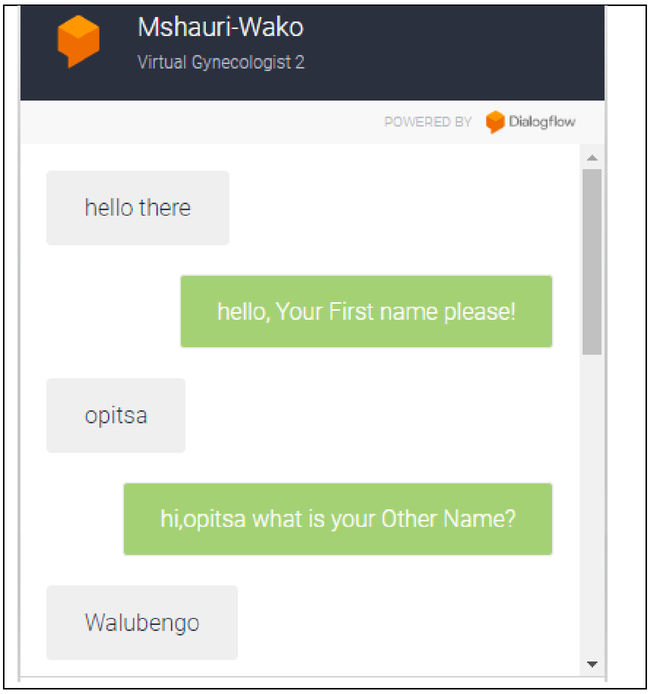
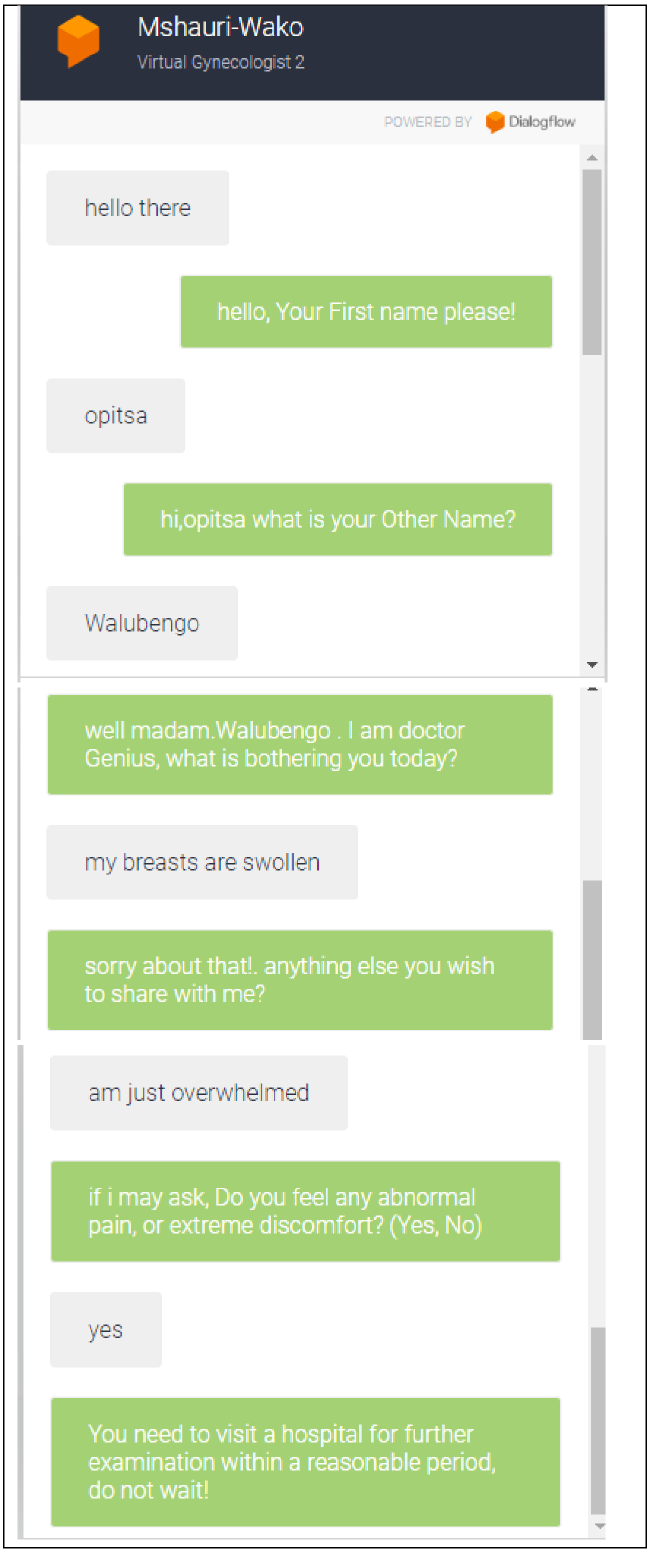


Figure B7.1 Sample perspective one conversations for both web and Facebook environments.




Mshauri-Wako
Virtual Gynecologist 2
POWERED BY Dialogflow
hello there
hello, Your First name please!
opitsa
hi,opitsa what is your Other Name?
Walubengo
well madam.Walubengo . I am doctor Genius, what is bothering you today?
my breasts are swollen
sorry about that!. anything else you wish to share with me?
am just overwhelmed
if i may ask, Do you feel any abnormal pain, or extreme discomfort? (Yes, No)
yes
You need to visit a hospital for further examination within a reasonable period, do not wait!

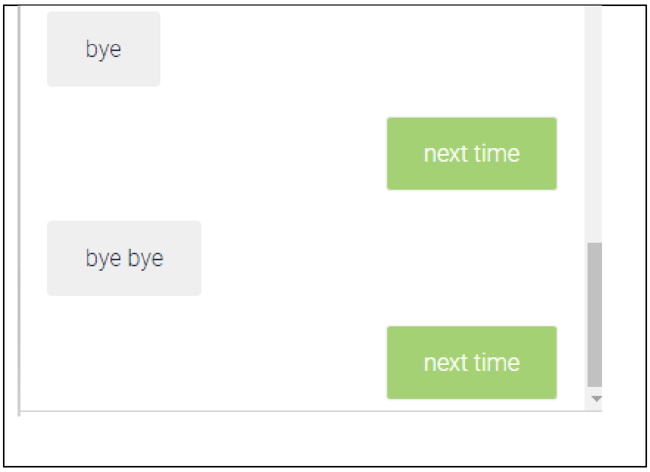


Figure B7.2 Sample perspective 2 conversation from web deployment.

APPENDIX 7C: SAMPLE AGENT CODE SNIPPET


```
# -*- coding: utf-8 -*-
{
  "responseId": "671adcf8-adba-4505-a4f7-de08947c2cf3-2a4c0c5e",
  "queryResult": {
    "queryText": "i have funny food cravings",
    "action": "agentAnswerQuery-general",
    "parameters": {},
    "allRequiredParamsPresent": true,
    "fulfillmentText": "Let me ask you, Do you feel any abnormal pain,
    "or extreme discomfort?  A Yes or No will be fine for an answer.",
    "fulfillmentMessages": [
      {
        "text": {
          "text": [
            "Let me ask you, Do you feel any abnormal pain, or extreme discomfort?
            "A Yes or No will be fine for an answer."
          ]         }       }
    ],
    "outputContexts": [
      {
        "name": "projects/testeragent-4fb7d/agent/sessions/
        "ba95a5b3-05d2-9ec8-d0d1-152ce333589f/contexts/agentanswerquery-followup",
        "lifespanCount": 2
      }    ],
    "intent": {
      "name": "projects/testeragent-4fb7d/agent/intents/
      "b8994eff-d945-483a-9d38-81f30081f198","displayName": "agentAnswerQuery"
    },
    "intentDetectionConfidence": 0.8492052,
    "languageCode": "en"
  }
```


Figure C7.1 MA_ agent answer user query

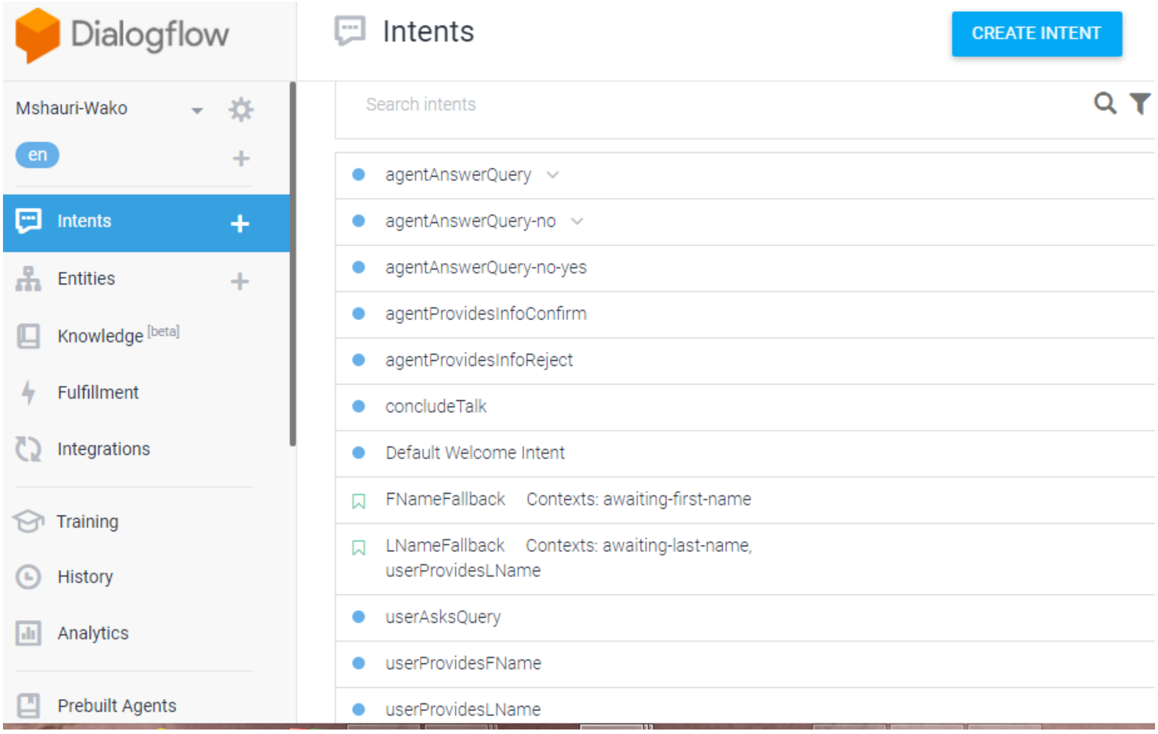


Figure D7.1. Sample Intents for the Master Agent.

Figure D7.1. shows a sample intents belonging to the master agent, during design time. At run time, the intents serve as intentions from the user or system to do something. Figure D7.2. shows a sample context belonging to the master agent, during design time. At run time, the context serves as a pointer referring to a specific subject of discussion.

The study suggests that the Figures D7.1. and D7.2. to be used for intuition purposes only. Otherwise the semantics of the full functionality has been provided in the JSON files, in appendices 7c.

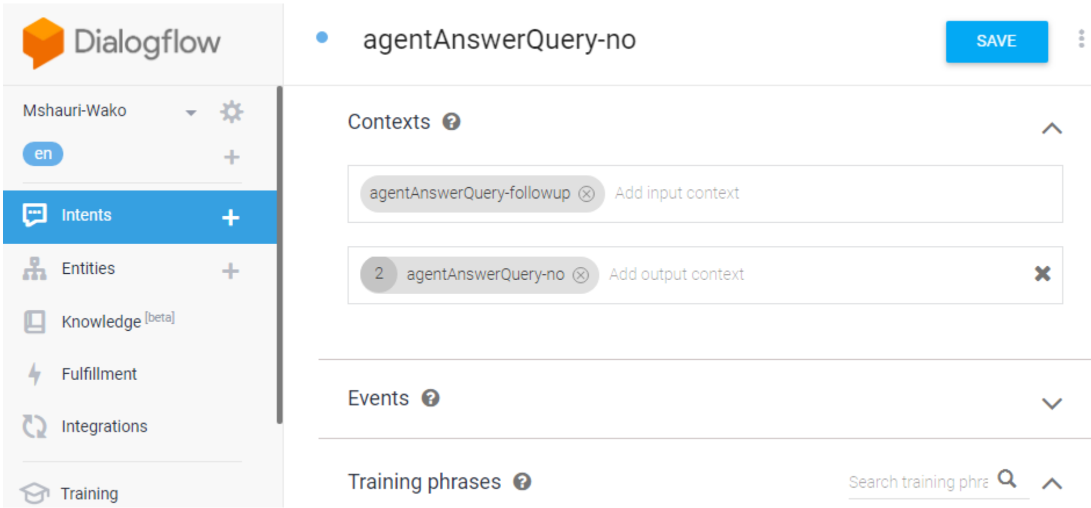


Figure D7.2. Definining context for a specific intent

APPENDIX 8: SAMPLE CONVERSATION

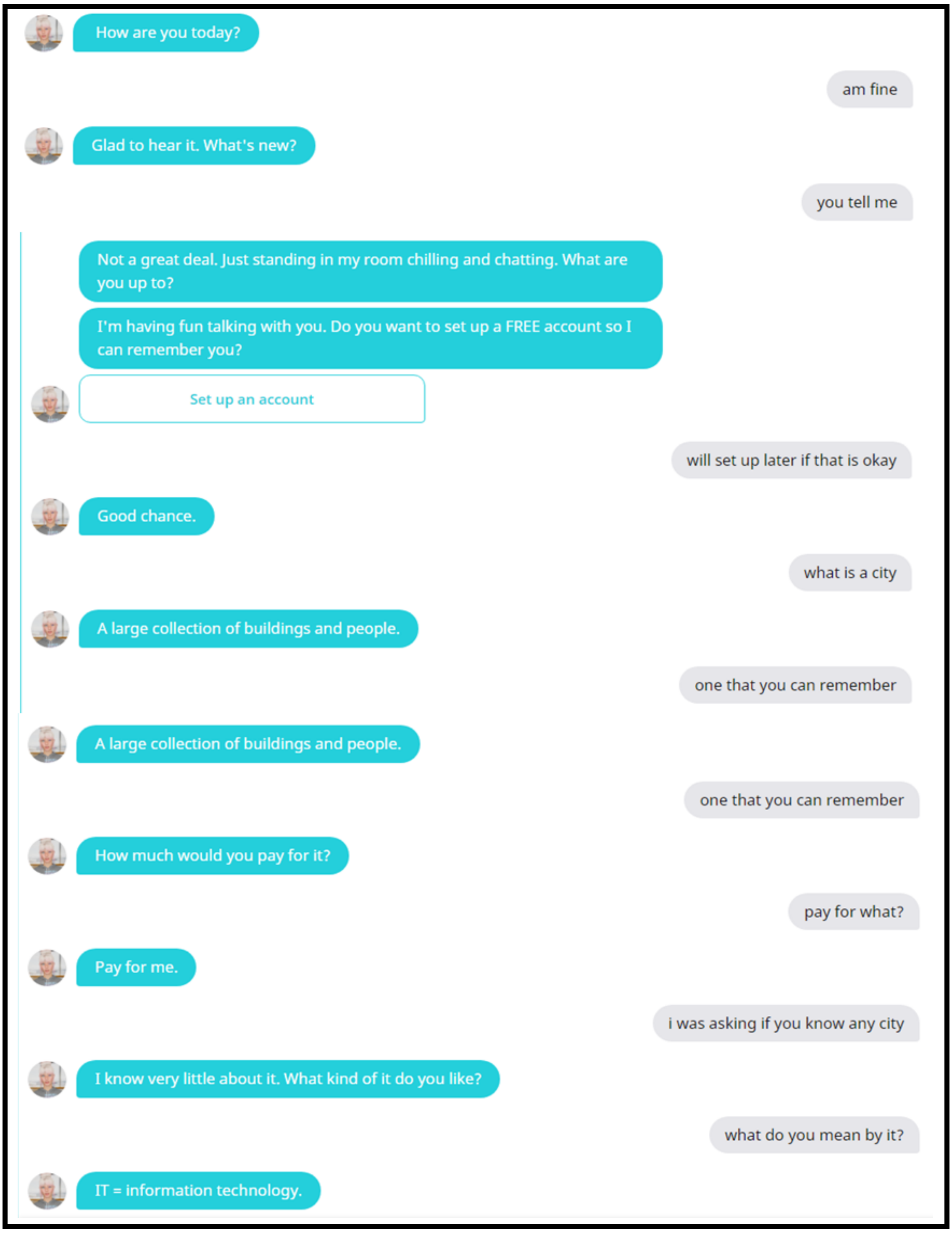


Figure A8.1. A human-agent dialogue during the process of Information inquiry, for a non-task-oriented dialogue. This shows the ambiguity in identifying "advancement in a conversation". In this case, it is not possible to tell whether there was the advancement or not; - a reality for most CODS. It is rather cumbersome for CODS to confirm and or deny the possibility of "advancing conversation".

APPENDIX 8A SCHEME FOR RESULTS INTERPRETATION

kappa statistic	
score	Interpretation
> 0.80	almost perfect
0.61 - 0.80	substantial
0.41 - 0.60	moderate
0.21 - 0.40	fair
0.01 - 0.20	slight
< 0	poor
machine learning is involved	
> 0.75	excellent
>0.5 < 0.75	good
> 0.4 < 0.5	fair
< 0.40	poor

Table A8.2 Acceptable kappa statistic by Landis [77] and Fleiss [78].

Appendix 8b Knowledge Source And Survey Url

Knowledge Source

https://americanpregnancy.org/
https://ehealthcaresolutions.com/author/ehs/
https://fhchc.org/

Survey URL
https://www.qsurvey.qa/home/en#/response/DOF58AHYJK

APPENDIX 9 : IMPORTANT CODE SNIPET.

```python
def list_intents(project_id):
    import dialogflow_v2 as dialogflow
    intents_client = dialogflow.IntentsClient()

    parent = intents_client.project_agent_path(project_id)
    intents = intents_client.list_intents(parent)
    for intent in intents:
        print('=' * 20)
        print('Intent name: {}'.format(intent.name))
        print('Intent display_name: {}'.format(intent.display_name))
        print('Action: {}\n'.format(intent.action))
        print('Root followup intent: {}'.format(
            intent.root_followup_intent_name))
        print('Parent followup intent: {}\n'.format(
           intent.parent_followup_intent_name))
        print('Input contexts:')
        for input_context_name in intent.input_context_names:
            print('\tName: {}'.format(input_context_name))
        print('Output contexts:')
        for output_context in intent.output_contexts:
            print('\tName: {}'.format(output_context.name))
```

Figure A9.1. Code snippet for automatic training using Dialogflow Auto ML models

```json
{
  "classificationAnnotation": {
    "displayName": "greetings"
  },
  "textContent": "inline_text",
  "dataItemResourceLabels": {
    "aiplatform.googleapis.com/ml_use": "training|test|validation"
  } },
{
  "classificationAnnotation": {
    "displayName": "User queries"
  },
  "textGcsUri": "gcs_uri_to_file",
  "dataItemResourceLabels": {
    "aiplatform.googleapis.com/ml_use": "training|test|validation"
  } },
{
  "classificationAnnotation": {
    "displayName": "System Response"
  },
  "textGcsUri": "gcs_uri_to_file",
  "dataItemResourceLabels": {
    "aiplatform.googleapis.com/ml_use": "training|test|validation"
  }
}
```

Figure A9.2. Code snippet for automatic training data continued.

```python
# Load and summarize the Pregnancy_info dataset
from pandas import read_csv
# Load dataset
# url = '//Pregnancy_info.csv'
# dataframe = read_csv(url, header=None)
dataframe = pd.read_csv("C:/Users/voke/python/ Pregnancy_info.csv")
# summarize shape
print(dataframe.shape)
```

Figure A9.3. Sample data pre processing

```python
# split into train test sets
from pandas import read_csv
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=1)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
# url = '/../ Pregnancy_info.csv' dataframe = read_csv(url, header=None)
dataframe = pd.read_csv("C:/Users/voke/python/ Pregnancy_info.csv")
```

Figure A9.4. Code snippet for showing dataset split. The code split the dataset so that 70 percent is used to train the model and 30 percent is used to evaluate it.

APPENDIX 10: COVER LETTER

21/6/2018
Kevin Mugoye Sindu,
Email: keymug2002@gmail.com
Department of Computer Science,
School of Computing and Informatics,
Maseno University,
P.O. BOX 333-40100
Kisumu – Kenya.

Dear Respondent,

RE: VOLUNTARY PARTICIPATION IN RESEARCH STUDY

I am a PhD student at Maseno University, School of Computing and Informatics, Computer Science department, currently conducting a study. My research study focuses on the use of Conversational AI to develop some intelligent interface to help minimize the thinking load from users interacting with an application or machine. The purpose of this study is to demonstrate that we can deploy new architecture and methods to equip conversational interfaces with ability to have a more natural conversation with a human user. This interface, in the form of an AI Chatbot, will be deployed on maternal healthcare domain which will also serve as a testbed for the system.

We need to evaluate the system using data from users other than development and supervisory team. The findings from the study will be used as proof and to justify the efficacy of the architecture from a holistic approach.

To achieve this objective, we invite your voluntary participation in using or testing the system prototype and filling the questionnaire, all will be provided online. As a participant, you have the rights to participate or withdraw from the study at any time. In case any physical or physiological harm that may affect your participation occurs during the study, necessary steps will be undertaken to mitigate the risk or injury. To safeguard your privacy, the information you provide during the study will be treated as confidential and will only be used for the purpose of this study. We appreciate your sacrifice for making this study a success.

Thank you.

Yours faithfully,
Kevin Mugoye Sindu
(Admin No: PHD/CI/00032/2015)

APPENDIX 11: TEST INSTRUCTIONS

Dear tester

Mshauri-Wako Chatbot is an example of smart information retrieval system.
Mshauri-wako is a virtual maternal health care advisor, the Chatbot prototype is equipped with maternal health information and can dispense information and offer advice on the same. It is equipped with pregnancy related information. Kindly note that at the moment the Bot has partial knowledge on one area and is not a substitute to a human medical expert.

The objective of this test is to determine whether the Chatbot can conduct a conversation that is meant to achieve a goal. The Bot is expected to take the conversation deeper, (either to get more information or to provide more information). It tries to mimic a maternal health expert.

You should figure out how to work with it. HOWEVER, you are required to try at least 4 (four) times before giving your feedback. It can enable a user to understand what is happening in the journey of pregnancy.
Use normal, simple English as you interact.

Assumptions

1) A user wants to know whether she is pregnant or not
2) A user may want to know information about a particular stage of pregnancy
3) A user may want to know any other stage that might be of interest to her.
4) A user should have knowledge of basic pregnancy symptoms.

APPENDIX 12 : RESEARCH PERMIT AND AUTHORIZATION LETTER

NATIONAL COMMISSION FOR SCIENCE, TECHNOLOGY AND INNOVATION

Telephone:+254-20-2213471, 2241349,3310571,2219420
Fax:+254-20-318245,318249
Email: dg@nacosti.go.ke
Website : www.nacosti.go.ke
When replying please quote

NACOSTI, Upper Kabete
Off Waiyaki Way
P.O. Box 30623-00100
NAIROBI-KENYA

Ref: No. **NACOSTI/P/19/79694/29516**

Date: **25th April 2019**

Kevin Mugoye Sindu
Maseno University
Private Bag
MASENO.

RE: RESEARCH AUTHORIZATION

Following your application for authority to carry out research on ***"A multi-agent reinforcement learning based approach to advancing conversation in task oriented dialog systems."*** I am pleased to inform you that you have been authorized to undertake research in **Selected Counties** for the period ending **25th April, 2020.**

You are advised to report to **the County Commissioners and the County Directors of Education, Selected Counties** before embarking on the research project.

Kindly note that, as an applicant who has been licensed under the Science, Technology and Innovation Act, 2013 to conduct research in Kenya, you shall deposit **a copy** of the final research report to the Commission within **one year** of completion. The soft copy of the same should be submitted through the Online Research Information System.

GODFREY P. KALERWA MSc., MBA, MKIM
FOR: DIRECTOR-GENERAL/CEO

Copy to:

The County Commissioners
Selected Counties.

The County Directors of Education
Selected Counties.