2012

# Improved Intention Discovery with Classified Emotions in A Modified POMDP

Sivaraman Sriram
*University of Windsor*

# Improved Intention Discovery with Classified Emotions in A Modified POMDP-Based Dialogue System

**by**

**Sivaraman Sriram**

A Thesis
Submitted to the Faculty of Graduate Studies
through Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada

2012

# Improved Intention Discovery with Classified Emotions in A Modified POMDP-Based Dialogue System

by

**Sivaraman Sriram**

**APPROVED BY:**

Dr. Kenneth Cramer
Department of Psychology

Dr. Scott Goodwin
School of Computer Science

Dr. Xiaobu Yuan, Advisor
School of Computer Science

Dr. Subir Bandhyopadhyay, Chair of Defense
School of Computer Science

## DECLARATION OF ORIGINALITY

# ABSTRACT

Emotions are one of the most proactive topics in psychology, a basis of forceful conversation and divergence from the earliest philosophers and other thinkers to the present day. Human emotion classification using different machine learning techniques is an active area of research over the last decade. This investigation discusses a new approach for virtual agents to better understand and interact with the user. Our research focuses on deducing the belief state of a user who interacts with a single agent using recognized emotions from the text/speech based input. We built a customized decision tree with six primary states of emotions being recognized from different sets of inputs. The belief state at each given instance of time slice is inferred by drawing a belief network using the different sets of emotions and calculating state of belief using a POMDP (Partially Observable Markov Decision Process) based solver. Hence the existing POMDP model is customized in order to incorporate emotion as observations for finding the possible user intentions. This helps to overcome the limitations of the present methods to better recognize the belief state. As well, the new approach allows us to analyze human emotional behaviour in indefinite environments and helps to generate an effective interaction between the human and the computer.

## DEDICATION

To my dear friends and family

## ACKNOWLEDGEMENTS

I would like to take this opportunity to thank my supervisor Dr. Xiaobu Yuan for his encouragement and support in presenting this piece of work. Deep appreciation and gratitude goes to him for contributing his suggestions and ideas during my research. This survey will clearly explain the goals, current state of art and emerging trends associated with the inference of the belief state of a human user interacting with a software agent using a modified POMDP approach. The thesis will bring about a clear picture of the existing techniques involved in providing classification process, which leads to an explicit discussion about the advantages and drawbacks involved in those techniques.

I would also like to thank my thesis committee members Dr. Scott Goodwin, Dr. Ken Cramer and Dr. Subir Bandhyopadhyay for spending their valuable time providing feedback about thesis throughout my proposal and defence.

Finally, I would like to thank my friends/colleagues Dr. Yehia Kotb and Mr. Rajaprabhu Dhanapal for their moral support and valuable suggestions.

## TABLE OF CONTENTS

DECLARATION OF ORIGINALITY ........ iii

ABSTRACT ........ iv

DEDICATION ........ v

ACKNOWLEDGEMENTS ........ vi

LIST OF FIGURES ........ x

LIST OF TABLES ........ xiii

**CHAPTER**

**I INTRODUCTION**

1.1 Big Picture ........ 1
1.2 Problem Specification ........ 3
1.3 Motivation ........ 4
1.4 Contribution ........ 5
1.5 Consolidation of Thesis ........ 5

**II PRECURSORY**

2.1 Human Emotions and Classification ........ 6
2.2 Multimodal Dialogue Management ........ 7
2.2.1 Input ........ 7
2.2.2 Fusion ........ 8
2.2.3 Dialogue Manager and General Knowledge ........ 9
2.2.4 Fission ........ 10
2.2.5 Output ........ 11
2.3 Agents, Belief States and User Intentions ........ 11

**III BACKGROUND WORK**

3.1 Emotion Recognition from Different Input Types ........ 14
3.2 The Flat POMDP Model ........ 17
3.3 Information State Space Approach ........ 20
3.4 Factored POMDP Approach ........ 24
3.5 Hollnagel's Contextual Control Model ........ 26
3.6 Modified Partially Observable Markov Decision Process ........ 28

3.7 A Historical Information Approach of POMDP based Dialogue Management ........29
3.8 Conclusion ........32

IV THE PROPOSED WORK

4.1 Shortcomings of the Current Techniques ........33
4.2 The New Proposed Method ........35
4.2.1 Overview of Proposed Architecture ........36
4.3 Emotion Recognition using Customized Decision Tree ........39
4.3.1 Introduction to Decision Tree Model ........39
4.3.2 Customized Decision Tree Algorithm ........41
4.3.3 Intensity Calculation ........43
4.3.4 Senti-Strengh Analysis ........43
4.3.5 Algorithm ........44
4.3.5.1 Emotion Strength Update ........45
4.3.5.2 Homogeneity Problem ........46
4.4 Inducing Emotions to Modified POMDP ........48
4.4.1 Rewards ........49
4.4.2 Confidence Score Evaluation ........50
4.5 Conclusion ........54

V EXPERIMENTATION AND RESULTS

5.1 Implementation Setup for Emotion Recognition ........55
5.1.1 Intensity Evaluation ........56
5.1.2 Senti-Strength Analysis ........57
5.1.3 Secondary Emotional Dataset ........59
5.1.4 Customized Decision Tree Implementation ........61
5.1.5 Customized Decision Tree Evaluation ........63
5.1.6 Validation of CDT ........66
5.1.7 Quantitative Analysis ........68
5.2 Implementation of Modified POMDP with Emotions ........70
5.2.1 POMDP File Specification ........72
5.2.2 Generating Policy File ........76
5.2.3 Test Case 1 ........79
5.2.4 Test Case 2 ........81
5.2.5 Test Case 3 ........84
5.3 Results and Discussion ........87
5.3.1 Analysis of Emotion Recognition ........89
5.4 Conclusion ........90

**VI CONCLUSION & FUTURE WORKS……….……………………………91**

**REFERENCES..................................................................................................93**

**VITA AUCTORIS ..........................................................................................102**

# LIST OF FIGURES

1.1.1 Human-Computer Interaction Diagram ....................................................................2

2.1.1 Customized Tree with Emotional Nodes (Primary and Secondary) .........................7

2.2.1 Multimodal Dialogue Manager Flow Chart .............................................................9

2.3.1 Overall State for a Given Time Slice.......................................................................13

3.1.1 Emotion Recognition Process in Hybrid Keyword Approach ................................15

3.1.2 Split Keyword based Emotion Recognition System................................................16

3.2.1 POMDP Model ........................................................................................................19

3.2.2 Transition of States in POMDP Model with Action and Observation .....................20

3.3.1 FSM based Dialogue Management Example ..........................................................22

3.3.2 Frame based Approach Example ............................................................................23

3.5.1 Flow of the Four Contextual Control Modes .........................................................28

3.7.1 Modified POMDP Approach ...................................................................................32

4.2.1.1 Proposed POMDP Transition Method with Emotions as Observation.................38

4.2.1.2 Proposed Architecture ..........................................................................................39

4.3.1.1 Decision Tree for the 5-coin Puzzle .....................................................................40

4.3.2.1 Overall Emotion Recognition Process..................................................................42

4.4.2.1 Flow Chart Representation of Proposed Work ....................................................52

4.4.2.2 Pseudo Code for New Approach...........................................................................53

5.1.1.1 Intensity and Binary Emotion Calculation............................................................56

5.1.2.1 Text Input to Senti-Strength.....58

5.1.2.3 Positive and Negative Strength Output.....58

5.1.3.1 Secondary Dataset Generation.....59

5.1.3.2 Extraction of Data from two Different Sources.....60

5.1.4.1 C# Application for Experimenting CDT.....62

5.1.5.1 Anger Emotion Class with Mean and RMS.....63

5.1.5.2 Disgust Emotion Class with Mean and RMS.....64

5.1.5.3 Joy Emotion Class with Mean and RMS.....64

5.1.5.4 Fear Emotion Class with Mean and RMS.....65

5.1.6.1 True Positive Ratio Graph for 10 Subsets of Training Sets.....67

5.1.6.2 True Positive Ratio Graph for 10 Subsets of Test Sets.....67

5.1.7.1 Pizza Order Application for Experimentation.....69

5.2.1 User Interface of User Simulator (POMDP mock up).....70

5.2.2 Overall Implementation Process of the Proposed Approach.....71

5.2.1.1 Tony Cassandra File Objects.....72

5.2.1.2 Tony Cassandra File Objects with 8 States.....74

5.2.1.3 Tony Cassandra File Objects with 21 States.....75

5.2.2.1 Generating Policy from POMDPSOL Binary File.....76

5.2.2.2 Terminating to Create the Output Policy File.....77

5.2.2.3 Policy Output File.....78

5.2.3.1 Test Case 1 in POMDP Mock up.....79

5.2.3.2 Test Case 1 POMDP Simulation........................................................................80

5.2.4.1 Test Case 2 in POMDP Mock up......................................................................81

5.2.4.2 Test Case 2 POMDP Simulation........................................................................82

5.2.5.1 Test Case 3 in POMDP Mock up......................................................................83

5.2.5.3 The 10 runs for the Output File..........................................................................84

5.3.2 POMDP Values with Multiple Scenarios ...........................................................88

## LIST OF TABLES

4.3.5.1 Possible Emotions Prediction Table ......................................................46

4.3.5.2.1 CDT Homogeneity Algorithm ..........................................................47

5.1.1.2 Intensity Values Metric (partial) ........................................................57

5.1.5.5 Predicted Unknown Emotion (CDT) .....................................................59

5.1.6.3 Confusion Matrix on Test Set ..............................................................68

5.2.3.3 Test Case 1 Results ...............................................................................80

5.2.5.2 Test Case 3 Results ...............................................................................84

5.3.1 Comparison Results ..................................................................................87

5.3.1.1 CDT Accuracy Comparison Table..........................................................89

# CHAPTER I

# INTRODUCTION

## 1.1 Big Picture

The recent buzz word in human-computer interaction research is the recognition of human emotion. Emotion classification has its own usage in making the conversation between the human and the computer even more interactive and pragmatic; and more importantly the system can respond appropriately to his/her emotional feelings [1]. According to Myers [2], emotions are “the complex psycho-physiological experience of an individual’s state of mind as interacting with biochemical (internal) and environmental (external) influences”. As well he stated that “in humans, emotions fundamentally involve physiological arousal, expressive behaviors and conscious experience”. There are many applications in which this can have a major contribution such as a human-like robot that interacts with the user, 3D animations, and graphical designs in cinema.

Recognition of emotions can be done using different media of inputs. Some kinds of media are speech, gestures and textual inputs and so on. Each medium has its own variety of recognition and both advantages and disadvantages. For example, considering the textual medium, the chat text must either contain an emotional keyword or we could use machine learning approaches, e.g., Knowledge-based Artificial Neural Network (ANN), to mine emotions from the text [3]. Basic emotions such as neutral, fear, anger, sadness, surprise, disgust and joy can also be deduced using facial recognition approach [4]. In comparison, another source of information is emotional keywords because textual emotions and direct emotional words such as ‘HAPPY’ are helpful for emotion recognition. In addition, emotional dataset labelled with semantic and syntax tags have been used for facial recognition [4].

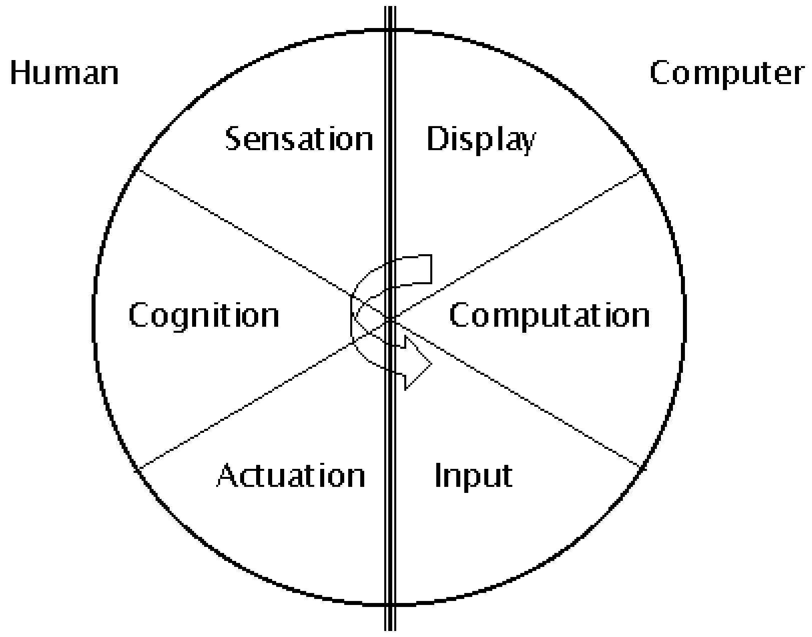


Figure 1.1.1: Human-Computer Interaction Diagram [5]

The belief state is a probability value that refers to the state of the environment at a given timestamp. Inferring the belief state in a dialogue conversation using any formatted medium (maybe even combined) is necessary to make the conversation look more like human-human interaction which is the bench mark for human-computer interaction. Figure 1.1.1 shows the important features of human-computer interaction as a circle separated into six equal pie wedges [5]. The belief state deduction of a human who interacts with a computer can be added up to a multimodal dialogue manager in order to make the conversation more appropriate and natural. For example, knowing the state of belief of a customer buying an online product would be helpful in giving him the suitable offers with respect to the product. A multimodal dialogue system is a computational device or agent that engages in interaction with other human, uses human language in some form such as speech, text, or gesture and typically engages with human such interaction across multiple turns or sentences [6]. A detailed explanation about the multimodal dialogue manager is given in Section 2.2 of this thesis.

## 1.2 Problem Specification

Numerous methods of dialogue management have been proposed in the last two decades, including the traditional approach based upon Finite State Machine (FSM) and the more recent approach built upon the prevalent Partially Observed Markov Decision Process (POMDP) model. Finite state machine based approach is suitable to the tasks that are finely organized and is not flexible. Frame based approach uses one frame at a given instance to observe and record the information and is much better with respect to flexibility when compared to FSM based method [7]. These approaches still do not have a comprehensive solution to solve decision or uncertainty based problems. Hence Bayes network and/or Markov decision process based approaches are used to solve some uncertainties and probability based problems to some point but still have disadvantages such as imperfections in solving reflection uncertainties [4]. Even though POMDP is the existing popular approach, it still has its own difficulties that have to be dealt with to improve its performance. Regardless of its identified problem of scalability, this approach establishes undisputable benefits in handling the input of uncertainty over other known approaches. However, the current method has a domain knowledge base in the planning process, and uses not only the current, but also the previous belief state for the determination of actions; the approach still did not quite completely understand user's intention as it does not consider the emotional states of the user. In an outlook, the POMDP-based approach only models the user and maintains the knowledge at the control level of a process but does not consider the fact that the interacting user has more than just conditional decisions. This thesis introduces human emotions as an observation and also considers historical beliefs as factors into the planning process of the POMDP to determine the actions and possible states, consequently improves the approach stated above.

## 1.3 Motivation

The Partially Observable Markov Decision Process (POMDP) is a mathematical model for sequential decision problems in partially observable environments. The term 'partial' means that the state of the world cannot be sensed directly [6]. The acceptance of embodied agents with a dialogue manager such as the POMDP depends on the naturalness and smoothness when interacting with users. Information about user emotion helps to discover the user intention, but has not been used in even the currently most advanced dialogue manager, due to the omission of history information in POMDP models. Thus the current techniques in solving a problem using POMDP based approach has its own limitations which gives an opportunity to come up with a better approach to solve the problem by modifying the existing methodology. This thesis conducts an investigation on this particular method and tries solving the challenges associated with the prevailing POMDP based dialogue manager.

## 1.4 Contribution

In this thesis, two main contributions have been made. First, the emotions of the interacting user is estimated using a customized decision tree algorithm from using different input methods such as text and speech forms. Second, with drawing the conclusion that a POMDP based approach drops some noteworthy evidence in terms of the historical information space theory with additional observations, the modified POMDP-based dialogue management approach is proposed to handle the uncertainties in the belief state. As well, it helps to improve the precision of determining user's intention using the recognized emotions. Experiments under different scenarios are conducted to evaluate the suggested technique. The results obtained from different experimentation support this notion and validate that the proposed approach accomplishes the expected results.

## 1.5 Consolidation of Thesis

The remainder of this thesis is structured as follows: Chapter 2 briefly reviews the general multimodal dialogue management and its components. Chapter 3 presents a survey of prior works that are related to the proposed method in two different parts, i.e., emotion recognition and user intention discovery. Chapter 4 states the research objectives and shows the need of user's emotion as an observation in discovering intention and belief update. Finally chapter 5 describes the implementation of the dialogue management with modified POMDP incorporating the emotions in the prediction of belief state. It also demonstrates experimental results.

# CHAPTER II

# PRECURSORY

## 2.1 Human Emotions and Classification

Emotions are a basis of forceful conversation and divergence according to the earliest philosophers and other thinkers to the present day. Myers [2] states that the emotions arise from physiological and psychological excitement. This statement can be explained simply by saying that the happiness comes from smiling and the sadness comes from crying. Classification of human emotions has become an increasingly important element for affect-sensitive human-computer interaction. The components of emotion are distinguished on the basis of physiological or psychological factors and include emotion faces, elicitors, and neural processes. Primarily, emotions are classified into six basic types: joy, disgust, surprise, anger, sadness and fear [8] as shown in Figure 2.1.1. We employ neutral emotion in computer science to represent a no-emotion state [9]. This is because the system involves textual/spoken data in predicting emotions in which if a particular sentence does not provide enough information for estimation, then the concluding result would be a neutral emotion.

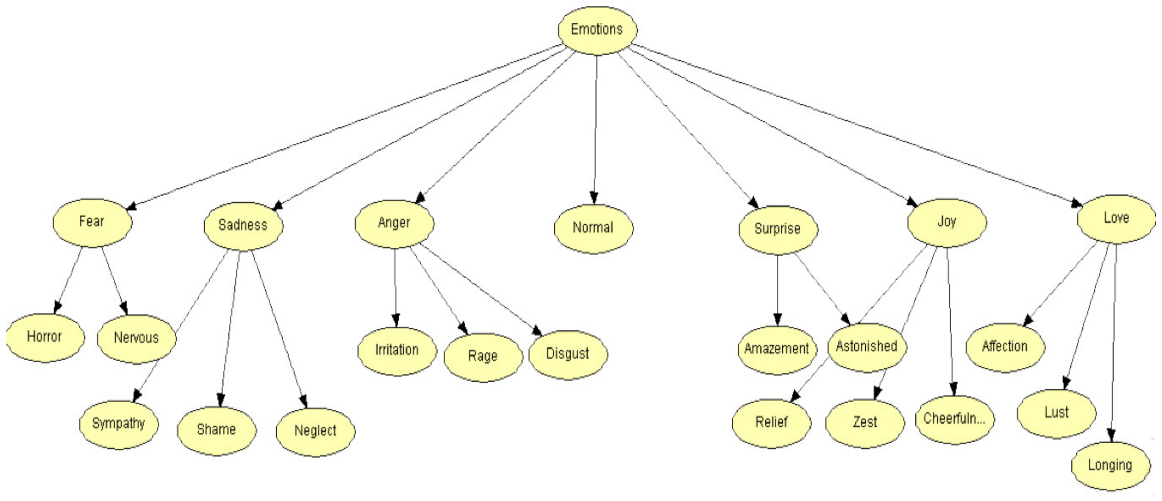


Figure 2.1.1: Customized Tree with Emotional Nodes (Primary and Secondary) [8]

The secondary state of emotions is described using a tree structure, which is inherited from their respective primitive types. These emotions can be recognized and classified using various sets of input levels such as textual conversations, speech based recognition, facial recognition and dynamic gesture recognition capturing the human body movements.

## 2.2 Multimodal Dialogue Management

### 2.2.1 Input

Multimodal systems have various modalities with respect to input modes such as speech, facial expressions, gestures, gazes etc. We have two kinds of inputs, namely, active input modes and passive input modes. Active input modes are explicit commands of user's intentions such as speech. Passive input modes refer to the user's behaviour that is recognized by the computer like facial expressions and manual gestures. They involve user inputs with no explicit commands and monitors passively.

### 2.2.2 Fusion

The input nodes give the information extracted from the user for extraction, recognition and integration. In this module the agent processes the information and assigns a semantic representation which is eventually sent to the dialogue manager. Fusion is classified into two categories: feature-level fusion and semantic-level fusion. The first one is a method for fusing low-level feature information from parallel input signals within a multimodal architecture like feature extraction. The second one is a method for integrating semantic information derived from parallel input modes in a multimodal architecture like action recognition (speech, gestures, so on). Low-level fusion is a sensory fusion in which sensory data from the sources results in better information as an output, while semantic-level fusion in the dialogue manager needs a knowledge source which already has a collective familiarity about the input .

### 2.2.3 Dialogue Manager and General Knowledge

A dialogue manager is the core component of a dialogue system. It maintains the history of the dialogue, adopts certain dialogue strategies, retrieves the content stored in files or databases, and decides on the necessary response to the user. The dialogue manager creates and updates an Information State corresponding to a notion of dialogue context. The dialogue moves have the effect of updating information states and moves can be initiated. The term "dialogue context" can be viewed as the totality of conditions that influence the understanding and the generation of communicative behaviour.

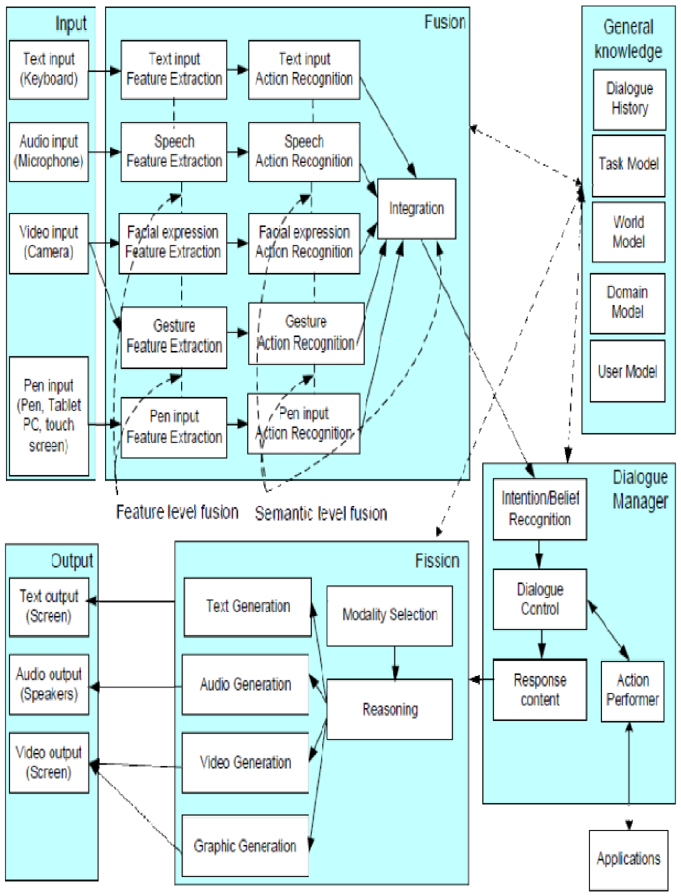


Figure 2.2.1: Multimodal Dialogue Manager Flow Chart [7]

The main tasks are updating the dialogue context on the basis of interpreted communication, deciding what content to express next and when to express it and interfacing with task/domain processing. A number of general knowledge sources is used by the dialogue manager such as Fusion and Fission which forms the overall *dialogue* history, task model, world model, domain model and user model. The dialogue model is a set of historical record of the spoken/ textual dialogue so far in terms of the entities. This representation provides a basis for conceptual coherence. Task model is a representation of the information to be gathered in the dialogue. This record, often referred to as a form, template, or status graph, is used to determine what information has not yet been acquired. World model contains general background information that supports any commonsense reasoning required by the system. Domain model uses the specific information about the domain; user model may contain relatively stable information about the user that is relevant to the dialogue such as the user's age, gender, and preferences as well as information that changes over the course of the dialogue, such as the user's goals, beliefs, and intentions. Figure 2.2.1 shows the process flow of the multimodal dialogue manager.

### 2.2.4 Fission

Fission is a process of understanding an abstract message from the dialogue manager in which the information is in machine understandable format. The tasks of a fission module are composed of three categories. The first is for content selection and structuring, in which the presented content must be selected and arranged into an overall structure. The second is for modality selection, which determines the optimal modalities based on the current situation of the environment. For example, when the user device has a limited display and memory, the output can be presented as the graphic form such as a sequence of icons. The third category is for output coordination, which is responsible for the coordination of all the output channels that the resulting output forms a coherent presentation.

### 2.2.5 Output

Many ways of output modalities can be used to present the information content from the previous module such as: speech, text, graphics, avatars etc. Some of the common output combinations are: speech and text, speech, text and graphics, speech and gestures, graphics and avatar, text and graphics, speech, graphics and animation.

## 2.3 Agents, Belief States and User Intentions

In artificial intelligence, an intelligent agent is an autonomous entity which observes and acts upon an environment and directs its activity towards achieving goals (Goal Oriented Agents). Intelligent agents may also learn or use knowledge to achieve their goals. Agents are often classified into two categories according to the techniques they employ in their decision making: *reactive* agents base their next decision solely on their current sensory input; *planning* agents, on the other hand, take into account anticipated future developments, for instance as a result of their own actions to decide on the most favorable course of action. The software agent we are using in this research is reactive as well as a planning agent, which can be obtained using different styles of agent modeling.

The belief-desire-intention (BDI) model has come to be possibly one of the well known and studied models of practical reasoning agents. There are several reasons for its success, but perhaps the most compelling is that the BDI model combines a respectable philosophical model of human practical reasoning as originally developed by Bratman [11]. The explicit goals to achieve and events to handle are the desires of agents. A set of *plans* (*intentions*) is used to describe how agents achieve their *goals*. Each *plan* describes how to achieve a *goal* under varying *environments* (*belief*). A set of data called *belief* describes the *state* of the *environment*.

Stanford's philosophical encyclopaedia states that [12] "Belief state or state of belief is considered as the propositional attitude of the user interacting with an agent". According to Brown [13], a state of belief is a primary bearer of truth-values in every given instant of time and propositions being its object. It is also conditioned that the belief state of an organism (human in our case) depends on two factors. The first condition is that the belief state of the user at a given time frame should supervene on his/her intrinsic properties. Secondly, a belief state must be a state of the user that depends only on those facts that are relevant to what he/she believes. Thus a proposition at a given time slice depends on the total state and the current situation. If we could deduce this proposition which is nothing but the belief state at a given instance of time or a particular situation, then the positive sum of all proposition divided by the number of instances would present the overall belief state.  Figure 2.3.1 shows an overall state diagram for a given instant of time.

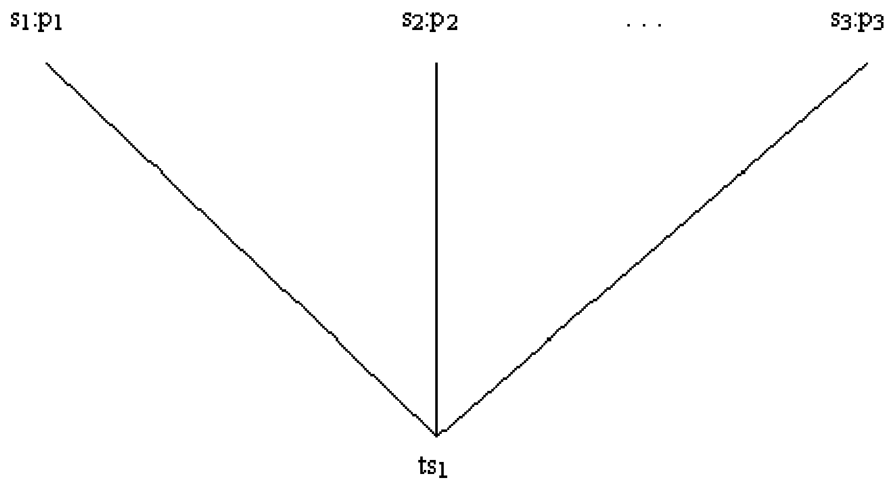


Figure 2.3.1: Overall State for a Given Time Slice [13]

In order to assist the user effectively to search on what they need and also learn the necessary methodology, the computer needs to understand the user's intention. Thus the belief state that we specify above is the intentional probability of the user. The user's intention can be classified into two different levels: *Action* intention and *Semantic* intention. Action intentions are lower level, such as mouse click, keyboard typing and other basic actions performed on a computer. Semantic intentions correspond to what the user wants to achieve at high level, which may involve several basic actions on a computer to accomplish it. In this thesis we concentrate more on semantic level intentions.

# CHAPTER III

# BACKGROUND WORK

## 3.1 Emotion Recognition from Different Input Types

This chapter is dedicated for highlighting major contributions from previous work in the field of emotion recognition. Seol [3] proposed a method for recognizing emotions from a textual data by using knowledge-based Artificial Neural network, which was hybridized with the traditional keywords based search to improve the efficiency which is shown in Figure 3.1.1.

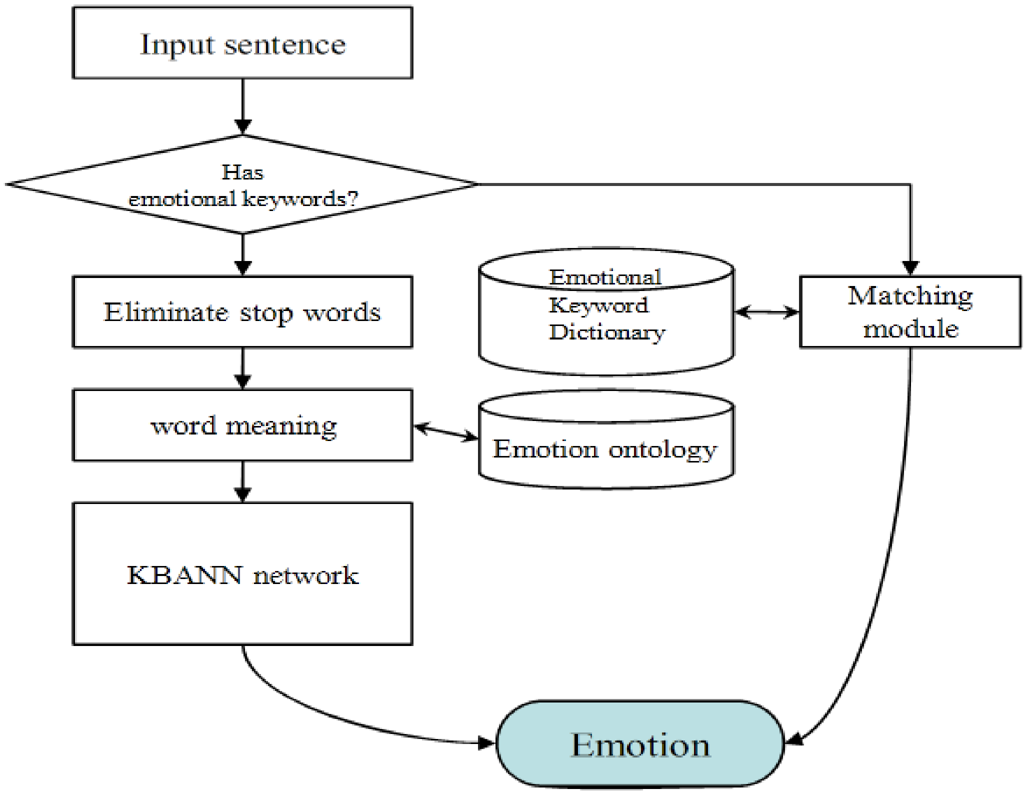


Figure 3.1.1: Emotion Recognition Process in Hybrid Keyword Approach [3]

Chunling [8] gave a preliminary method in estimating the emotion in a text based chat system using the split keywords based searching which has client-server architecture as shown in Figure 3.1.2.

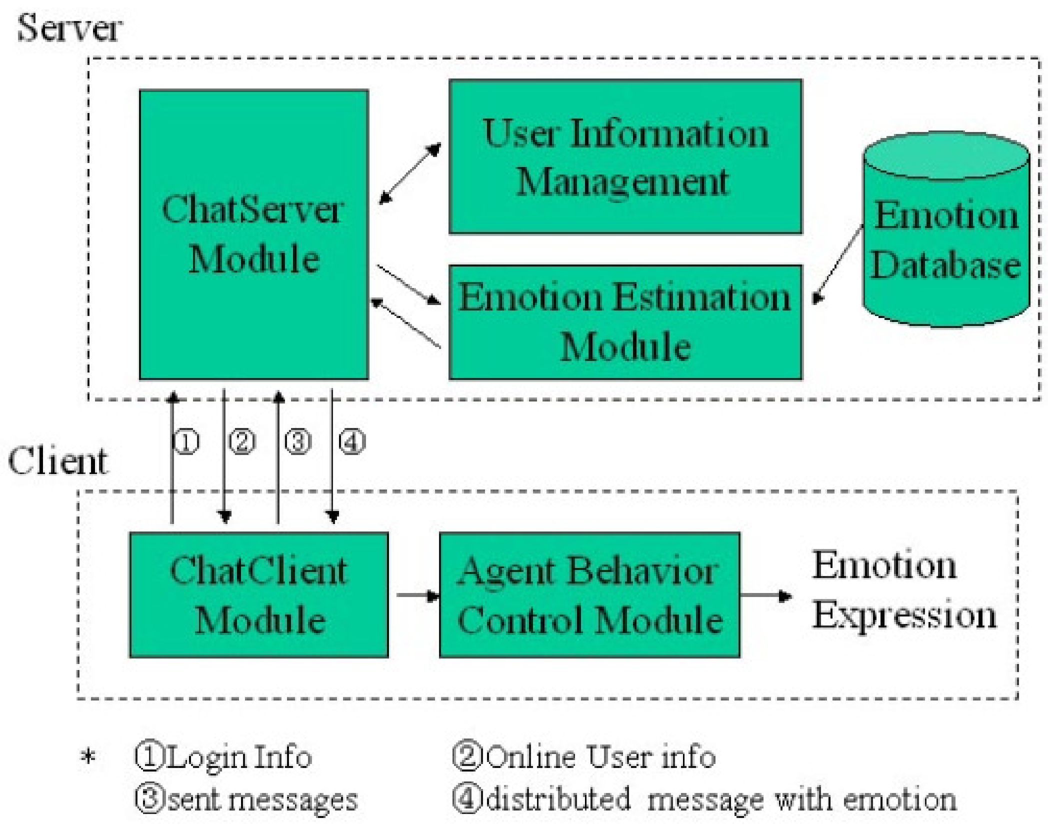


Figure 3.1.2: Split Keyword based Emotion Recognition System [8]

Yu- Lu [14] designed a system which uses a semantic role labeling system that finds the all-important constituent in each paragraph and then identifies the emotion using web mining. His research is useful in a textual emotional mining but the method utilizes a lot of memory and also uses a web based search like Google. Wu [15] proposed an approach for automatic recognition of emotions from the text in which emotion generation rules (EGRs) are manually deduced to represent the conditions for generating emotion. Based on the EGRs, the emotional state of each sentence can be represented as a sequence of semantic labels (SLs) and attributes (ATTs); SLs are defined as the domain-independent features, while ATTs are domain-dependent.

Wong [16] presented a novel approach for recognizing facial emotion in order to further detect human suspicious behaviors. Instead of relying on relative poor representation of facial features in a flat vector form, the approach utilizes a format of tree structures with Gabor feature representations to present a facial emotional state. Cheng [17] proposed an automatic facial expression recognition system. This system develops a semantic-based learning algorithm using the analytical hierarchy process (AHP), which is created to bridge the gap between low-level visual features and high-level semantics. Jamshidnajad [18] presented a facial expression recognition model using fuzzy techniques in order to further detect human behaviors in the e-business. A fuzzy clustering model is proposed to classify images after extracting the features that are used as inputs into the classification system. The outcome of this model is one of the preselected emotional categories within the given image set.

Castellano [19] proposed an approach for the recognition of acted emotional states based on the analysis of body movement and gesture expressivity showing that distinct emotions are often associated with different qualities of body movement. He used non-propositional movement qualities to infer emotions and propose a method for the analysis of emotional behaviour based on both direct classification of time series and a model that provided indicators describing the dynamics of expressive motion cues. Egges [20] described a generic model for personality, mood and emotion simulation for conversational virtual humans. He presented a generic model for updating the parameters related to emotional behaviour and gave a linear implementation of the update mechanism. He also explored how existing theories for appraisal can be integrated into the framework to form a prototype in combination with a dialogue system and a talking head with synchronized speech and facial expressions. Devillers [21] reported on three studies: the first concerns the use of multi-level annotations including emotion tags for diagnosis of the dialogue state; the second investigates automatic emotion detection using linguistic information; and the third reports on two perceptual tests for identifying emotions as well as the prosodic and textual cues which signal them and lastly propose a new set of emotions.

## 3.2 The Flat POMDP Model

The POMDP models an agent taking a sequence of actions under uncertainty to maximize its reward. Formally it is specified as a tuple(S, A, O, T, Z, R, $\gamma$), where S is a set of states, A is a set of actions and *O* is a set of observations. In each time step, the agent lies in some state $s$ ε S; it takes some action $a$ ε A and moves from s to a new state $s'$. Due to the uncertainty in action, the end state s′ is modeled as a conditional probability function $T(s, a, s') = p\ (s'|\ s, a)$, which gives the probability of which the agent lies in s′, after taking action ‘a’ in state s. The agent then makes an observation to gather information on its state. Due to the uncertainty in observation, the observation result o which belongs to O is again modeled as a conditional probability function $Z(s, a, o) = p(o|\ s, a)$ [6] [22].

In each step, the agent receives a real-valued reward R(s, $a_m$), if it takes action 'a' in state s, and the goal of the agent is to maximize its expected total reward by choosing a suitable sequence of actions. For infinite-horizon POMDP, the sequence of actions has infinite length. It also specifies a discount factor $\gamma$ belonging to [0, 1) so that the total reward is finite and the problem is well-defined. In this case, the expected total reward is given by E [P(t) $\gamma$(t) R($s_t$, $a_t$)] where $s_t$ and $a_t$ denote the agent's state and action at time t. A policy $\pi$ induces a value function V(b) that specifies the expected total reward of executing policy $\pi$ starting from b. It is known that V*, the value function associated with the optimal policy $\pi*$, can be approximated arbitrarily closely by a convex, piecewise-linear function V(b) = max $\in$ L($\alpha$.b), where L is a finite set of vectors called $\alpha$-vectors, b is the discrete vector representation of a belief, and '$\alpha$.b' is the inner product of vectors $\alpha$ and b. Each $\alpha$-vector is associated with an action. The policy can be executed by selecting the action corresponding to the best $\alpha$-vector at the current belief. Therefore a policy can be represented by a set of $\alpha$-vectors. Figure 3.2.1 shows the flow of the POMDP model.

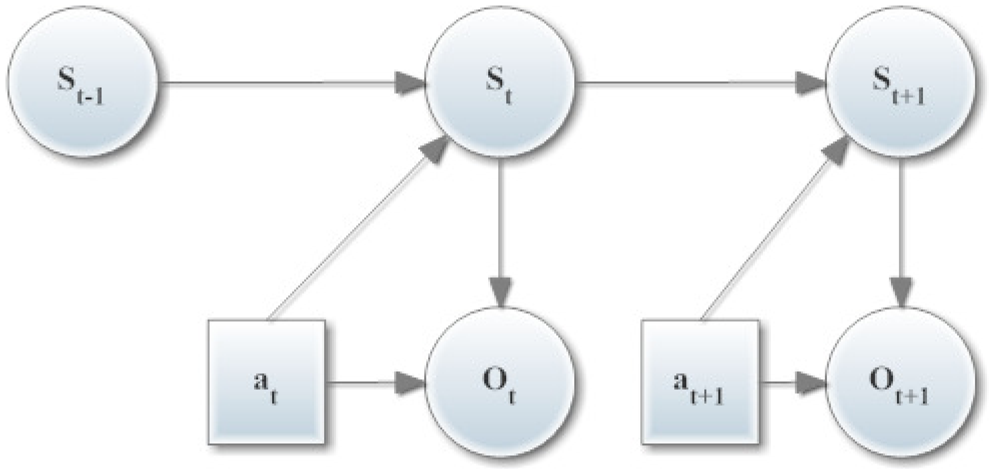


Figure 3.2.1: POMDP Model [23]

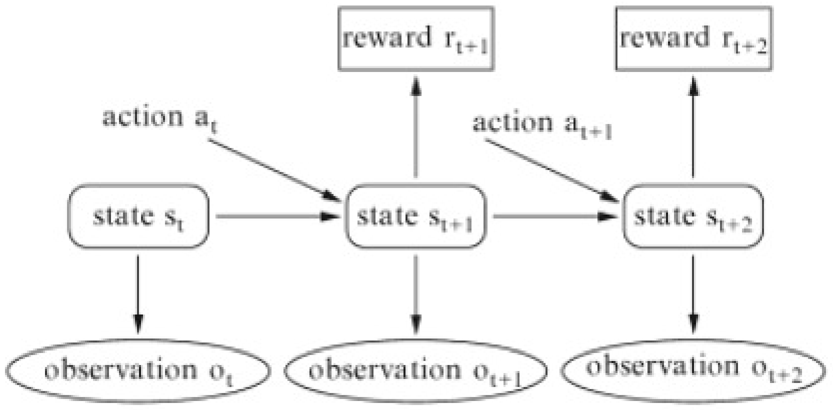


Figure 3.2.2: Transition of States in POMDP Model with Action and Observation

The state estimator component of a POMDP updates the belief state of the agent every time it executes an action. Given the belief state of the agent at time t as $b_t$, it gives an opportunity to compute the belief state at time t + 1, $b_{t+1}$, after a transition in the process where the agent occupies state S, executes an action *a* and perceives an observation *z* which is shown in Figure 3.2.2 [23]. The belief that the agent is in the resulting state s' is derived by:

$$\begin{aligned} b_{t+1} &= P(s'|z,a,b_t) \\ &= \frac{P(z\,|s',a,b_t)P(s'|a,b_t)}{P(z|a,\ b_t)} \\ &= \frac{P(z\,|s',a,b_t)\sum_{s\,\varepsilon\,S}\quad P(s'|a,b_t,s)P(s|a,b_t)}{P(z|a,\ b_t)} \\ &= \frac{P(z\,|s',a)\sum_{s\,\varepsilon\,S}\quad P(s'|s,a)P(s|a,b_t)}{P(z|a,\ b_t)} \\ &= \frac{O(s',a,z)\sum_{s\,\varepsilon\,S}\quad T(s,a,s')b_t(s)}{P(z|a,\ b_t)} \end{aligned}$$

In essence the above equation evaluates the probability of ending up in states given that the agent had a belief about its own state $b_t$, executed an action and perceived an observation z according to the predefined observation and transition functions of the POMDP, O() and T() respectively. The denominator $P(z \mid a, b_t)$, is a normalizing factor and is equal to the total probability of perceiving the observation z given the previous belief state of the agent and the action it executed:

$$P(z \mid a, b_t) = \sum_{s' \varepsilon S} \mathrm{P(z|s', a)P(s'|s, a)b_t(s)}$$

$$= \sum_{s' \varepsilon S} \mathrm{O(z, s', a)T(s, a, s')b_t(s)}$$

## 3.3 Information State Space Approach

A finite state model based dialogue system is one of the primitive dialogue management approaches. The system directs the user with prearranged questions designed by the developers to complete some task. Finite State Machine (FSM) based approach models the dialogue flow and task representation with nodules and edges. Each nodule in the model stands for the system utterance and the edges keep in touch to the user's answers which establish all possible paths through the set of connections. This approach is the most familiar and simple model. The framework collects one piece of information at a time. Before submitting all the information to knowledge base or database it will definitely confirm with the user. Both task model and dialogue model are inherent and they are programmed by a dialogue designer. Baekgaard [24] discussed this model and applied this approach in the Danish dialogue project. They have used a basic finite state set-up to model the dialogue flow for a repeated book club service. More details about the theory of this model are described in [25]. A dialogue model in the domain of the train ticket issuing system will be used in the following of this section to illustrate the various dialogue management approaches. In this domain, the ticket can be issued after both departure city and arrival city information obtained. Figure 3.3.1 illustrates the finite state machine based approach of dialogue management under above mentioned domain.

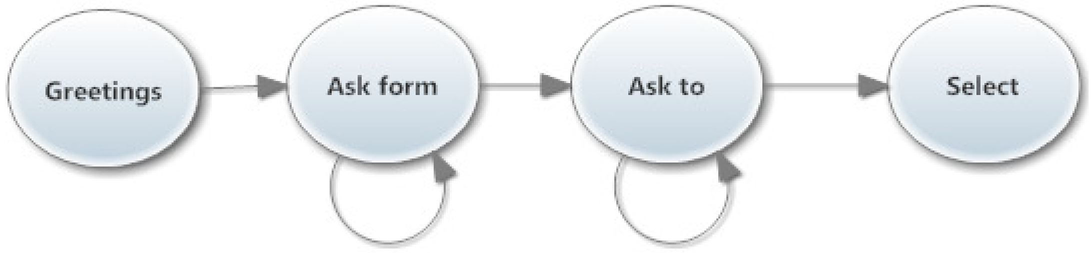


Figure 3.3.1: FSM based Dialogue Management Example [25]

McTear [26] pointed out that the palpable advantage of the finite state model approach is effortlessness and only suitable for the thought through task. The questions to be asked and their sequences are predetermined. In the whole dialogue session, the agent guides the user and constrains the layout of the user's answers. After each turn in the dialogue, the agent will explicitly interpret with the user what have been assumed presently. As there are so many restrictions within the dialogue, the agent does not require sophisticated knowledge applied such as natural language processing. The advantage of the finite state based approach meanwhile reflects the shortcomings including: it can only apply to the uncomplicated domain as it lacks flexibility in the dialogue management. During the dialogue, the user neither manipulates the dialogue nor brings in new dialogue subject. When more uncertainties are brought by the users or environment, the system can easily crash because of the unsuitable dialogue policy generated by the domain's expert and restricted stipulated script. In finite state based approach, the dialogue system is agent directed and only collects each section of information at every turn based on its current dialogue state. When the user introduces more information than the system necessitates at each dialogue state, there exists a problem. Finite state machine based approach neither realizes the manifold slots filling nor deals with the outmoded information brought by the user. As an annexe of finite state based model, frame-based model is developed to prevail over the lack of flexibility of the finite state machine based method. The frame-based approach is like a task of slot-satisfying where a slot is an encoded set of information that should be congregated by the agent. The dialogue is conducted to fill in the vacant slots. It also allows some amount of mixed-initiative and several slot fillings, which decides the dilemma within the approach. However, the conversation model is still programmed by a dialogue designer based on their familiarity and consideration. The frame based approach is illustrated in Figure 3.3.2[6]:

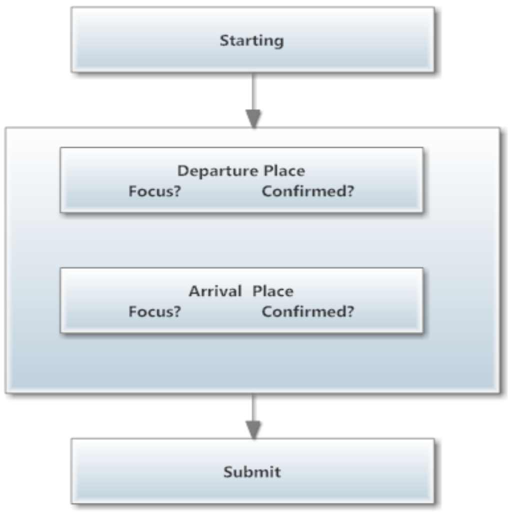


Figure 3.3.2: Frame based Approach Example

Ward and Pellom [27] used the analogous mechanism in their raconteur system, in which the next action of the agent is produced based on the current framework rather than stipulated script.  Jonsson [28] used information requirement forms under the sphere of influence of bus timetable information system. A more flexible frame based approach was projected by Goddeau [29] named '*E-form*' which has been applied in a spoken language boundary to a classified advertisements for used car database. He also abridged other variations of frame-based models which tolerate to deal with other complex dialogues. These variants include schemas that are used in the Carnegie Mellon Communicator system to represent more multifarious tasks [30] [31] and task structure graphs which provide a semantic structure and are used to determine the behaviour of the dialogue control as well as the language understanding component [32]. Type hierarchies are used to model the domain of a dialogue and as a basic for clarification questions [33]. Blackboard is used to manage contextual information applicable to dialogue manager such as history board, control board, presentation board, etc [6]. Frame based approach can comprehend the mixed imitative dialogue and put up with redundant information brought by the users. The sequence of the questions or the information to be gathered is not pre-fixed, which is based on the current context to generate next query to ask. However, McTear [26] summarized that the next step that is based only on the existing context is not enough. Hence he suggested a problematical domain in which the state of the world is dynamic and the knowledge level of the user is diverse which can not apply for the classical frame based approach.

## 3.4 Factored POMDP Approach

William [34] casted the spoken dialogue system as a factored POMDP to use this model as general framework for existing POMDP dialogue manager. In this model, the POMDP state variable $s \varepsilon S$ into three components such as: 1) the user's goal, $s_u \varepsilon S_u$; 2) the user's action $a_u \varepsilon A_u$; 3) history / state of the dialogue $s_d \varepsilon S_d$. Thus, the POMDP state 's' is given by the tuple$(s_u, a_u, S_d)$. As well, from the system's perspective, all those components are unobservable. The user's goal, $s_u$ gives the current goal or intention of the user. For example, user goal includes a complete travel schedule, a product the user would like to purchase or requesting information about it and so on. The user's action $a_u$, gives the user's most recent actual action. For example, identifying a place the user would like to travel, replying to yes/no question, or a null response indicating the user took no action. The dialogue history/state $S_d$, indicates any relevant history or state information. For example, if a particular slot has not been stated and if there are any ungrounded items then the dialogue designer might wish to penalize asking an open question. The POMDP action $a_m \varepsilon A_m$ is the action the machine takes in the dialogue such as greeting the user or asking a query. At each time stamp, the POMDP receives a single observation but it maintains a distribution over all possible user actions $a_u$. The factored POMDP is given by decomposing the POMDP transition function which is as follows:

$$P(s'|s, a_m) = P(s'_u, s'_d, a'_u|s_u, s_d, a_u, a_m)$$

$$= P(s'_u|s_u, s_d, a_u, a_m)P(a'_u|s'_u, s_u, s_d, a_u, a_m)P(s'_d|a'_u, s'_u, s_u, s_d, a_u, a_m)$$

The first term points out the user goal model. At each time step *t,* it is assumed that the user's goal depends on the previous goal and the machine action.

$$P(s'_u|s_u, s_d, a_u, a_m) = P(s'_u|s_u, a_m)$$

The second term is the user action model which indicates what action the user is likely to take at each time step *t.* It is assumed that the user's action depends on the current goal and preceding machine action.

$$P(a'_u|s'_u, s_u, s_d, a_u, a_m) = P(a'_u|s_u, a_m)$$

The third term is the dialogue model which indicates how the user and system actions affect the dialogue history. The current state or history of the dialogue depends on the previous history / state of the dialogue, user's action and system action. Thus, the transition function of POMDP is given by,

$$P(s'|s, a_m) = P(s'_u|s_u, a_m)\ |P(a'_u|s'_u, a_m)\ P(s'_d|a'_u, s_d, a_m)$$

The observation function of POMDP is given by,

$$P(o'|s', a_m) = P(o'|\ s'_u, s'_d, a'_u, a_m)$$

The confidence score and rewards are not specified as this model is associated with a particular user goal and designated objectives of the target system respectively. At each time *t,* the actions are selected depending on the belief state to maximize the cumulative long-term reward by substituting and simplifying the above equations. The belief state of the next state is given by,

$$b'(s'_u, s'_d, a'_u) =$$

$$k.P(o'|a'_u)P(a'_u|s'_u, a_m) \sum_{s_u \, \varepsilon \, S_u} P(s'_u|s_u, a_m) \sum_{s_d \, \varepsilon \, S_d} P(s'_d|a'_u, s_d, a_m) \sum_{a_u \, \varepsilon \, A_u} \mathrm{b}(\mathrm{s_u}, \mathrm{s_d}, \mathrm{a_u})$$

This model is tested with a simulated dialogue management problem in a travel domain in which the user is trying to buy a ticket to travel and compared the results with handcrafted policies and MDP baseline [23] [35]. The results showed that POMDP maintains a well formed distribution over user goals and in case of uncertainty. In particular, it reflects the true user goals. However, since this model assumes the flat listing of flat components, the spoken dialogue systems with hierarchical components may result in poor performance.

## 3.5 Hollnagel's Contextual Control Model (COCOM)

Hollnagel [36] initiated the contextual control model to assess team behaviour based on the time available. The main contribution was that the system determined the actions based on the context of the situation and available time. He classified team behaviour into four different modes:

- Scrambled Mode
- Opportunistic Mode
- Tactical Mode
- Strategic Mode

Hollnagel's COCOM was tested by Stanton, et al. [37] where they created 24 groups of people with 4 people in each group. There were 74 males and 22 females between 19 and 55 years of age. They made six groups of four people in each group to work on balancing a simulated gas network system. As an outcome of this experiment they reliably categorized the four control modes and showed that the progression between control modes conformed to a linear progression.

Shown in Figure 3.5.1 is the COCOM model, featuring the movements available between the different control modes. We have applied the same COCOM to the proposed model for creating a mock-up data and testing it with the real time POMDP based solver.

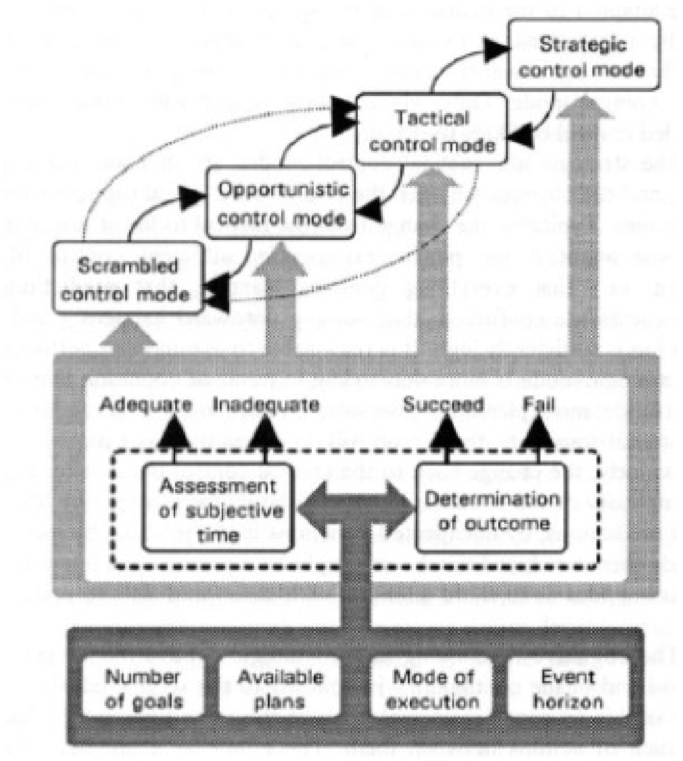


Figure 3.5.1: Flow of the Four Contextual Control Modes (COCOM) [36]

## 3.6 Modified Partially Observable Markov Decision Process

The Modified POMDP model was established to overcome the limitations of the existing POMDP approaches. According to Sabiha [7] the existing POMDP approaches considered the world to be static and always made their decisions based on the current belief state. It was also pointed out how the existing POMDP approaches ignored the dialogue history to make decisions effectively. Hence the author proposed a new POMDP based approach known as the Modified Partially Observable Markov Decision Process which shared Hollnagel's Contextual Control Model (COCOM) for decision making based on user input [36].

The dialogue manager toggles from one mode to another for processing user queries. For example, if the dialogue manager is not able to recognize the user request it switches to the scrambled mode where panicking occurs. The dialogue manager will attempt to move to tactical mode by giving the user a set of alternative options whereas other dialogue managers try to replicate the same question again and again until the user comes up with the right answer forming an infinite loop.

The dialogue manager attains the goal by changing over from one mode to another using forward planning. The lowest level of control will be the scrambled control mode where the system does not have a proper understanding of the user's queries and the most desirable level of control mode will be strategic mode where the system has a clear understanding of the user's queries. The system chooses the best action based on the context of the dialogue conversation and transitions between the modes depending on the dialogue states, current action, context of the situation and available time. The system provided evidence for being effective in handling the uncertainty caused by speech recognition errors and performed much better at handling conflicts in comparison to the existing POMDP approaches.

## 3.7 A Historical Information Approach of POMDP based Dialogue Management

The existing approaches of the dialogue management suffer from inflexibility during human-computer interaction as in FSM-based approach. They also lack the ability in handling any ambiguity as in frame-based and bayes-network approaches, and exhibit insufficiency when dealing with uncertainties as in the POMDP-based approach. To overcome the shortcomings while retaining the advantages of POMDP-based approaches, Bian [6] [38] proposed a modified planning strategy as illustrated below

$$\Pi_{new}: \ I_{k-1} \cup I_k n \rightarrow U$$

In the new approach, both $I_k$ and $I_{k-1}$ are still in the form of belief state, and state updating still uses the existing POMDP model as described. The addition of $I_{k-1}$ in the modified approach, however, introduces an important element to dialogue management, i.e., the history of belief state or the dynamics of belief state. Although the historical information of observations and actions is not maintained explicitly in $I_{k-1}$, the union $I_k$ and $I_{k-1}$ in the above equation diminishes the negative effect of Markov assumption and allows POMDP-based dialogue management to plan for actions with not only the current belief state but also the updated history before reaching the current state. The uncertainties that the original POMDP-based approaches fail to handle mainly arise from situations in which the user lacks knowledge in the domain or the user's goal cannot be fulfilled due to real-life constraints. In addition, dependency of factors in belief state also causes uncertainty.

The original POMDP-based approach is only able to resolve those uncertainties that are brought in by noise from observations, e.g., misinterpretation of words, and actions, e.g., misunderstanding of meaning. The dialogue system tries to "listen correctly" and to response appropriately to the user based on its state of belief [39]. By interrupting the planning process of POMDP-based dialogue management, a new component can be added to introduce a knowledge base with new rules and a database with practical constraints. Shown in Fig. 3.7.1 is the architecture for the modified POMDP-based dialogue management, in which the additional component interrupts the direct flow from b to π. As a realization of the new planning strategy, action alters the original action when there is an unexpected change from $I_{k-1}$ to $I_k$, or more accurately from the previous belief state to the current state [38]. The added component also skips the original planner π and makes direct contact with the user. At each stage of a dialogue, the new approach uses the domain knowledge and constraint database to help validating the change of belief state. The structure for the approach is shown in Figure 3.7.1. After an initial greeting, the system always updates the belief state with previous belief state, the current action, and the latest observation from the user.

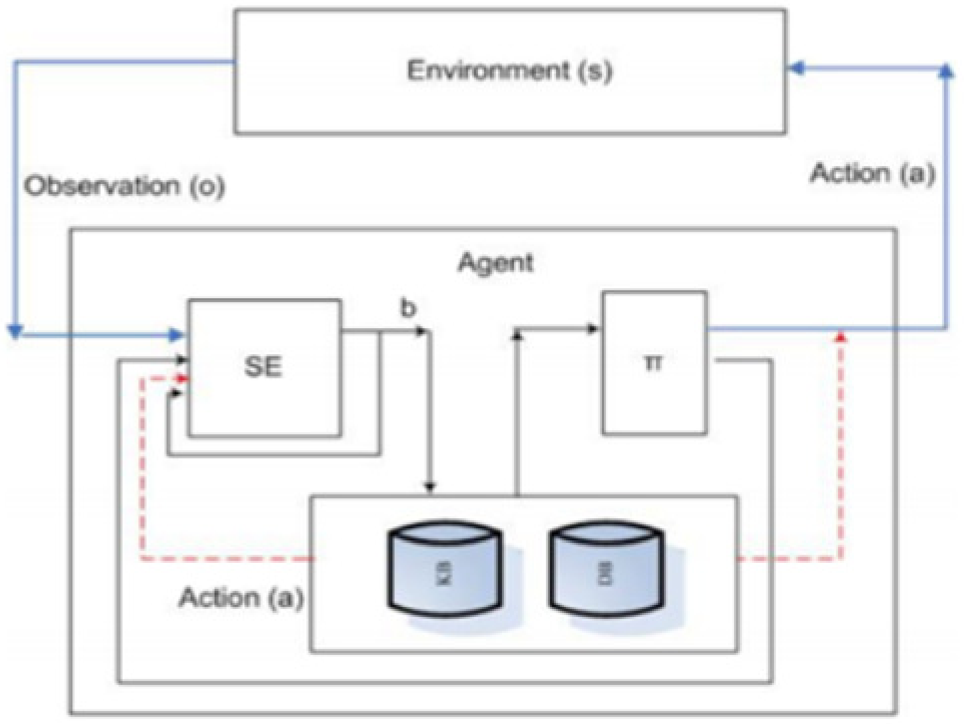


Figure 3.7.1: Modified POMDP Approach

If there is a failure in validation results a roll-back of belief state to the previous stage occurs, which means that the current state of the system will not be considered necessary to attain a particular user goal. Meanwhile, it triggers an explanation to the user and a question requesting for further information. This planning process is able to guide the user to reach a feasible goal that satisfies the need without causing conflicts.

## 3.8 Conclusion

In this chapter, we discussed briefly about different approaches of dialogue management, with emphasis on the more recent approaches based on POMDP and historical information space. While each of these approaches has its own advantages and drawbacks, there are still issues and limitations remaining unsolved. This thesis considers the historical information based POMDP model as the baseline and proposes a new method to overcome its disadvantages.

# CHAPTER IV

# PROPOSED WORK

Dialogue management is primarily delinquent under the influence of uncertainty. This chapter explains the proposed method of using the user's emotion to better predict the next possible belief state with the analysis of the history of user dialogues which is then used in the belief update. POMDP is the framework for agent planning under uncertainty and used for problem solving in dialogue management to make decision in the environment where the state of the agent is uncertain. POMDP can be used where the environment is noisy (one which has unnecessary information) and with the help of belief state the agent can reach a conclusion about the intension of the user. The problem with respect to dialogue management system needs the history to analyze the user intension in a better way and to reduce the uncertainty of the environment to the agent. This problem is solved with the help of the information space.

## 4.1 Shortcomings of the Current Techniques

The POMDP-based approach avoids the need to estimate system state by using a set of probability distribution over belief state in the planning process. Together with the action at the $k^{th}$ stage and the previous belief state, the system uses new observations to update the belief state and plans for action at the next stage. In the process, the states of both the system and the user are hidden in the information space. As defined earlier the flat model and the factored model, history information state is mapped to a probability distribution over the unknown system state. As this conversion is based on the Bayes filter theory, which in turn is under the Markov assumption, the POMDP-based approach plans for actions with only the current belief state, which is clearly illustrated in the background work section. Planning with POMDP models is better than all the other existing approaches as it does not rely on estimated system state, and is able to handle input uncertainty. However, the elimination of $I_o \cup I_1 \cup I_2 \cup I_3 \ldots \cup I_{k-1}$ from $I_{hist}$ makes it impossible to trace changes in belief state and to retrieve the historical information of observations and actions. In other words, belief state is a static probability distribution over the current system state only. As a consequence, the POMDP-based approach is unable to deal with uncertainty in belief state itself, which corresponds to uncertainty in either user's actions or the observation of user's actions. In another perspective, the POMDP-based dialogue management approach only models the user's goal or it can be considered as a user modeling rather than a task modeling or machine state modeling. When we are dealing with the observation and action uncertainties, the POMDP-based approach outperforms the other approaches. This advantage is even more obvious when the error rate of the input is high. Let us consider a situation when listening to the user's goal is not correct at the beginning. The task will finally end up with the failure although dialogue model listens correctly. Usually the dialogue systems make an assumption that the user can always answer the questions from the agent. However, in the real life condition, the user might have a lack of domain knowledge and could provide unreasonable information to the agent. This situation will be worse when the user cannot actually understand the question generated by the agent. If the dialogue management approaches model the user alone without its own domain knowledge level inference, the task cannot be achieved. In the process of the human-computer interaction, if the computer can appropriately influence the user and guide the user, the task is more probably to be achieved. These problems are overcome using the historical information space based approach for the POMDP but this method still does not consider the fact that the emotion of the user is important while deducing the next possible state. For example, the user's intention can be deduced to a particular scenario which could be suboptimal as it might be something which user does not want or might make him or her displeased or unsatisfied. This leads to the new proposed approach, which is to be explained in the next sub-section. In the proposed approach, the user's emotion is first inferred using a customized decision tree and these emotions are then perceived in the POMDP model as an observation including the action at every transition.

## 4.2 The New Proposed Method

This chapter gives a detailed explanation about the contribution of this thesis. After an overview of the proposed methodology, it discusses about the customized decision tree algorithm [1] and integration with spoken dialogue systems for decision making with user emotions inferred by this algorithm from known and unknown datasets. Details are then provided for a modified history space based ($I_{hist}$) POMDP approach, which becomes capable of predicting the user intentions when not only given user actions but also supplied with user emotions. To explain the customized decision tree process better, let us first give an overview of the standard decision tree model. Before we proceed further we would like to highlight the main contribution towards this thesis:

- *Classification of emotion of the given user's input (user action $U_a$) using customized decision tree algorithm.*
- *Introduce the inferred emotions to the spoken dialogue manager for decision making.*
- *Utilize the exiting modified POMDP approach which uses the historical information space to predict the overall state of belief or the user intention using not only the actions of the user but also the inferred emotions as an observation ($O_u + O_e$).*

### 4.2.1 Overview of Proposed Architecture

Analysis indicates that the compact of $I_{hist}$ of history information space into a resulting information space in a compressed form results in loss of important information [6]. The consequence is inflexibility for human-computer interaction as in the FSM based approach, incapable of handling any ambiguity as in the frame/Bayes/MDP based approaches, and insufficiency in dealing with uncertainties as in the POMDP-based approach [6] [7] [38]. To overcome the shortcomings while retaining the advantages of the modified POMDP-based approaches, this investigation helps to improve the intention discovery by adding emotions into the user observation (O) in the POMDP tuple(S, A, O, T, Z, R, $\gamma$).

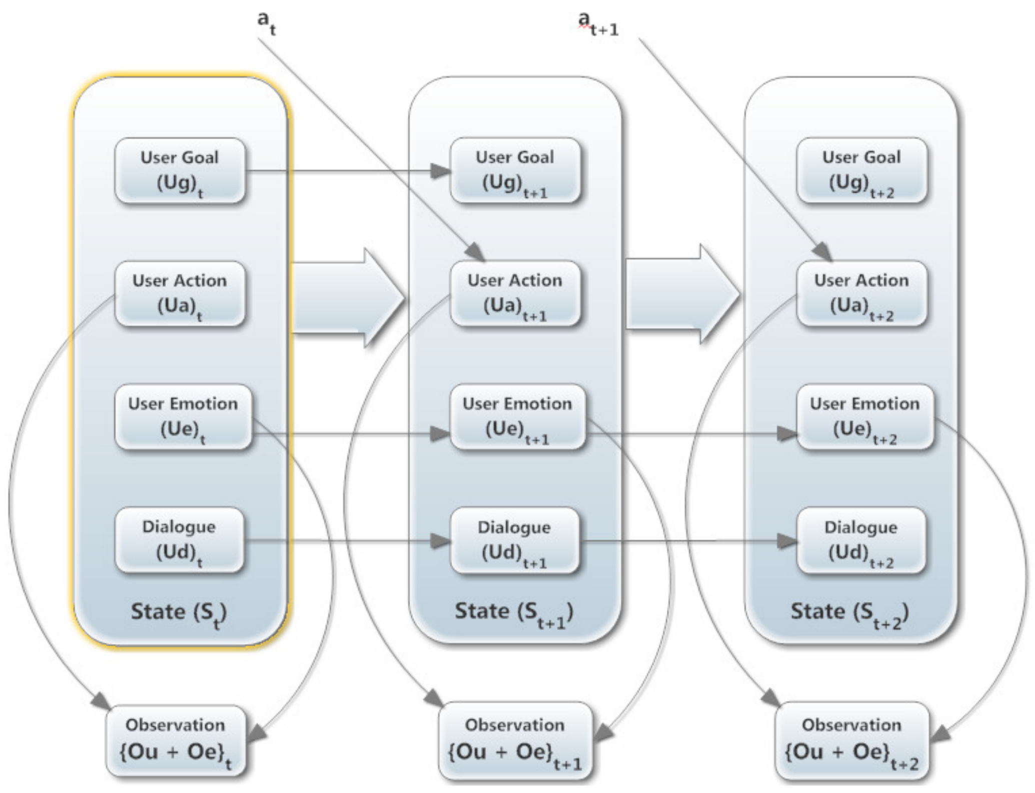


Figure 4.2.1.1: Proposed POMDP Transition method with emotions as Observation

The above state transition diagram explains the proposed dialogue model where the current POMDP approach has been extended with an impact of emotional component in it. Thereby, the State S is distributed into following four attributes: *User Goal, User Action, User Emotion* and *Dialogue State*. Thus an observation of the agent not only has the user action but also the emotional factor which helps improving the user intention prediction in the POMDP belief state analyser. As well, it is obvious that the user's action at every timestamp depends on system's action at the current timestamp, goals and emotions from the previous timestamp.

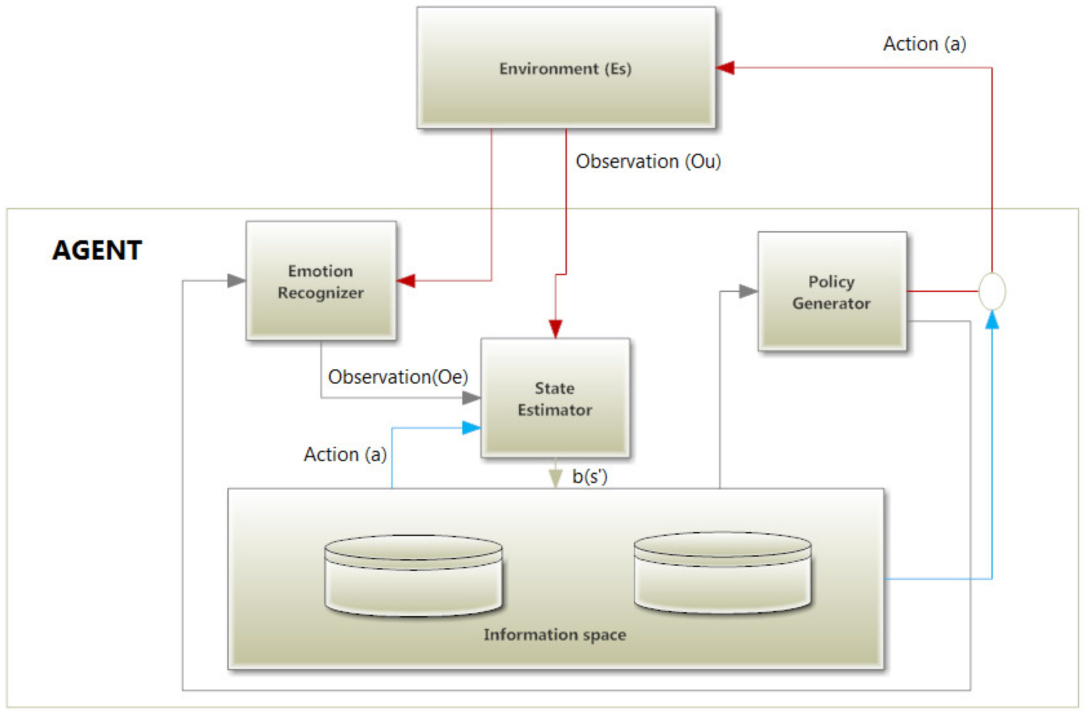


Figure 4.2.1.2 Proposed Architecture

Shown in Figure 4.2.1.2 is the architecture of the modified POMDP-based dialogue management in which the additional component interrupts the direct flow from *b* to policy (π). As a realization of the new planning strategy, the new policy is illustrated in the following equation.

$$\Pi_{new}: \; I_{k-1} \cup I_k n \rightarrow U$$

Action '*a*' alters the original action when there is an unexpected change from $I_{k-1}$ to $I_k$, or more accurately from the previous belief state to the current state. The added component also skips the original planner $\pi$ and makes direct contact with the user. Therefore, the two main aspects of this architecture are the emotion recognizer and giving it as an input to the state estimator in the dialogue manager. We would like to explain the emotion recognition process from textual and speech based data in the next sub-chapters and then would continue on how it infers the data into the dialogue model.

## 4.3 Emotion Recognition using Customized Decision Tree

In this section, it is explained in detail about how different classification techniques are used to predict the unknown user emotion using the given sets of data from the database and comparison of the same is also produced. The datasets are prepared to have more information to train the machine with different features about the data. Therefore this whole process contains three different steps namely: data import, feature extraction and addition (translate) and recognition. Before explaining Customized Decision Tree (CDT) it would be essential to understand the classical decision tree model [1].

### 4.3.1 Introduction to Decision Tree Model

The resource taken from North Western University [40] explains the decision tree in detail, which describes decision tree with the game 5-coin puzzle and explicitly shows the game trees associated. The diagram below explains the decision tree for the game of 5-coin puzzle:

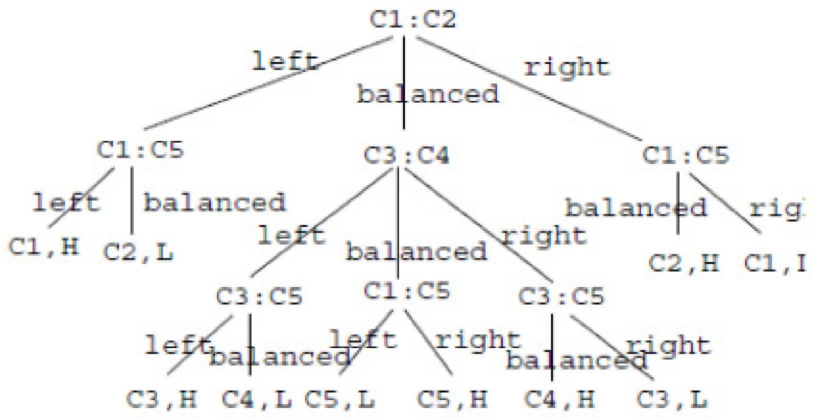


Figure 4.3.1.1: Decision Tree for the 5-coin Puzzle [40]

The Five-Coins Puzzle: In this puzzle we have five coins *C*1*,C*2*,C*3*,C*4*,C*5 that are identical in appearance, but one is either heavier or lighter that the others. The problem is to identify the bad coin and determine whether it is lighter or heavier using only a pan balance and comparing the weights of two piles of coins. The following discussion describes a solution to this problem.

First we compare the weights of *C*1 and *C*2. If *C*1 is heavier than *C*2 then we know that either *C*1 is the bad coin and is heavier, or *C*2 is the bad coin and it is lighter. By comparing *C*1 with any of the other coins, e.g. *C*5, we can determine whether the bad coin is *C*1 and is heavier (if *C*1 it is heavier than *C*5) or it is *C*2 and is lighter (if *C*1 has the same weight as *C*5). If *C*1 is lighter than *C*2 we proceed as before with “heavier” and “lighter” reversed. If *C*1 and *C*2 have the same weight we can try comparing *C*3 and *C*4 in a similar manner. If their weights are the same then we know that the bad coin is *C*5, and we can determine whether it is heavier or lighter by comparing it with *C*1. In each vertex of Figure 4.3.1.1 “*Ci* : *Cj*” means that we compare coins *Ci* and *Cj* by placing *Ci* on the left pan and *Cj* on the right pan of the balance, and each edge is labelled depending on what side of the balance is heavier. The terminal vertices are labelled with the bad coin and whether it is heavier (H) or lighter (L). The decision tree is optimal in the sense that in the worst case it uses three weightings, and there is no way to solve the problem with less than that—with two weightings we can get at most nine possible outcomes, which are insufficient to distinguish among ten combinations of 5 possible bad coins with the bad coin being heavier or lighter.

### 4.3.2 Customized Decision Tree Algorithm

A decision tree is a hierarchy based classifier in which each branch node represents an option between a number of alternatives, and each leaf node represents a decision [41]. The general algorithm for decision tree is as simple as a nested if-then-else structure. There are nine features used in this study of emotion recognition, namely strength of angry, strength of sad, strength of surprise, strength of fear, strength of disgust, strength of joy, intensity of emotion, positive and negative sentimental strength [1]. The proposed approach evaluates the RMS (Root Mean Square) and mean of each dimension in an emotion class first. We use RMS along with the arithmetic mean because it is useful when the emotion database has both positive and negative values.

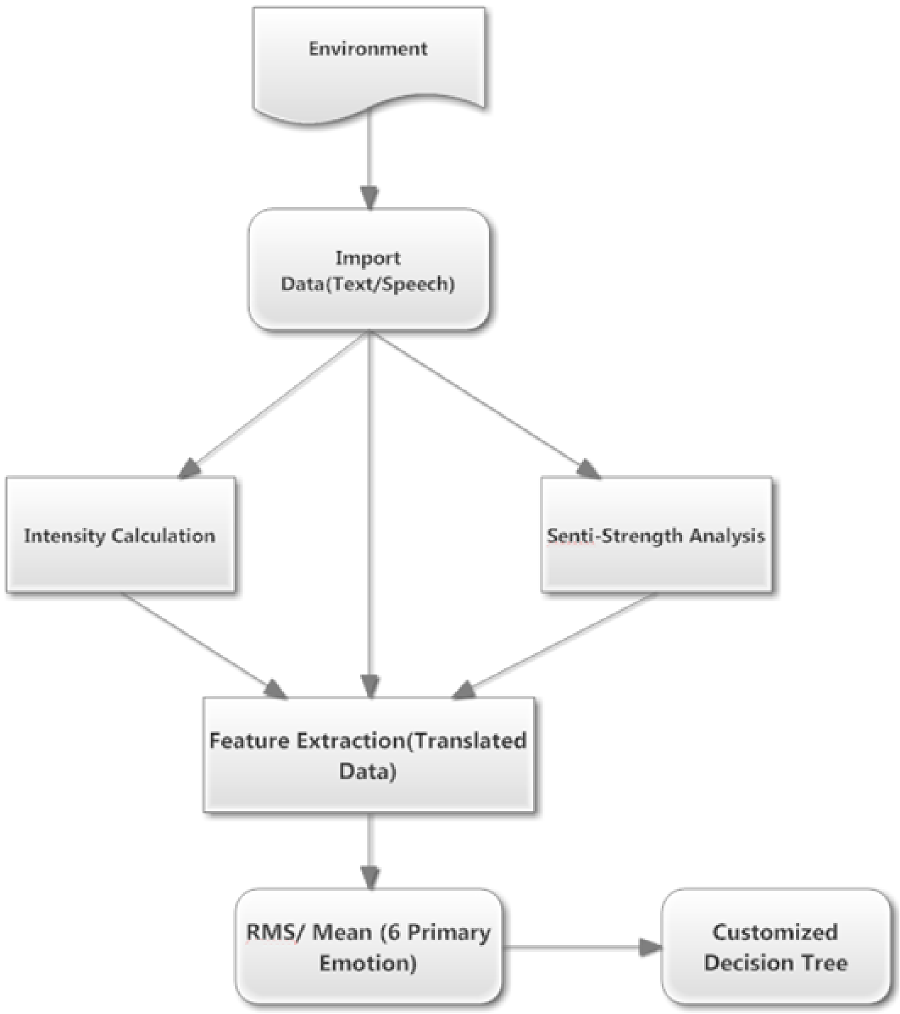


Figure 4.3.2.1 Overall Emotion Recognition Process [1]

The above diagram explains the overall architecture of emotion recognition process, where the necessary steps that have to be explained are namely, intensity calculation, senti-strength analysis and feature extraction module.

### 4.3.3 Intensity Calculation

The intensity that is evaluated in this investigation is nothing but the cognitive intensity of the user. It can be detailed as facets of thinking and dealing out information used in problem solving. The emotion texts are collected from known sets of data for evaluating the intensity. The details about the datasets that are used to train the system are explained in Section 5.1.1. Intensity of the emotion is analyzed with the formula as follows:

$$\text{Intensity(Iem)} = \frac{\text{Number of emotion keywords in the particular statement (Nkey)}}{\text{Total number of words in the statement (Ntotal)}}$$

Intensity of the emotion is calculated and updated in the database. This gives more information about the data for the machine to learn so that the efficiency of the learning should increase gradually.

### 4.3.4 Senti-Strengh Analysis

Positive and negative emotion strengths are updated to the database using the senti-strength software analysis. Senti-strength is a sentiment analysis (opinion mining) program designed to measure the *strength* of positive and negative sentiments in *short texts*, in which the language can be informal [42]. Fed with a set of short texts, it will allocate negative sentiment strength of -1 (least negative) to -5 (extremely negative) and positive sentiment strength of 1 (least positive) to 5 (extremely positive) to each one. Senti-strength is configured to analyze English language and is optimized for MySpace comments but can be modified for other languages and contexts by changing its configuration files. It should work reasonably well on any short English texts (i.e., a few sentences each) [1] [42]. Section 4.3.5.1 shows the emotional strength updated from this interface.

### 4.3.5 Algorithm

As stated earlier, a decision tree is a hierarchy based classifier in which each branch node corresponds to an option from a number of alternatives, and each leaf node represents a decision [41]. In this approach, first the RMS and the mean of each dimension in the emotion class are calculated. There are seven features used in this emotion study. These features structures the necessary dataset for the emotion recognition and are namely: strength of anger, strength of joy, strength of sad, strength of fear, strength of disgust, strength, strength of surprise and intensity of emotion.

For every feature which is represented by each column in the dataset, the value of mean and RMS are calculated. This will construct the Mean-RMS dataset with seven emotional strengths and intensity.

$$\text{Mean} = \frac{\text{Sum of all values under the dimension}}{\text{Total number of values}}$$

Thus the mean is calculated for every dimension and recorded. Secondly, for every dimension the average influence that can be evolved by every datum with respect to a given dimension can be calculated by finding the root mean square value for each dimension in both the classes. The RMS of a collection of $n$ values $\{x_1, x_2, x_3\ldots x_n\}$ is $x_{rms.}$

$$x_{\mathrm{rms}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} = \sqrt{\frac{x_1^2 + x_2^2 + \cdots + x_n^2}{n}}$$

Using the above formulae the RMS of each dimension under it is calculated, globally for all seven features. In this approach the whole dataset of 1000x7 is reduced to just two sets of arrays i.e. a mean array (mean [7] [7]) and RMS array (RMS [7] [7]) where the first dimension denotes emotion and the second dimension denotes the number of dimensions which is seven. The algorithm uses these two arrays as a metric in predicting the unknown classes of mixed dataset, then for each test data finds the distance metric of the dimensions with RMS[1][7], mean[1][7] and RMS[2][7], mean[2][7].

### 4.3.5.1 Emotion Strength Update

The emotion strength is updated for every feature in the Mean-RMS dataset. Positive and negative emotions are not used in the Mean-RMS calculation. Instead, the emotion strength is updated directly using the following customized rule.

| Negative rule | Positive rule | Possible emotions |
|---|---|---|
| $S_- = -1$ | $S_+ = 2$ | Joy, surprise |
| $S_- = -1$ | $S_+ > 2$ | Joy, Surprise |
| $S_- + S_+ = 0$ | | Do Nothing |
| $S_- = -2$ | $S_+ = 1$ | Sad, disgust |
| $S_- = -3$ | $S_+ < 3$ | Sad, Fear |
| $S_- = -4$ | $S_+ = 4$ | Sad, Fear, angry |
| $S_- = -5$ | $S_+ = 3$ | Angry, Sad, Fear |

Table 4.3.5.1: Possible Emotions Prediction Table [1]

The possible emotions with the value of positive strength and negative strength can be predicted. Based on the definition of the emotion strength the rules are framed to calculate the possible emotions and to update the points for a particular emotion observed in the statement.

### 4.3.5.2 Homogeneity Problem

The problem of homogeneity arises in the procedure of finding the emotion from the Mean-RMS dataset. This could be solved by finding the Euclidean distance measure between the unknown data and Mean-RMS value of each emotion. Table 4.3.5.2.1 explains this algorithm clearly with the entropy values for the datasets. The distance measure (DM) or the Euclidean distance is calculated between only two different emotion sets at a particular time. We calculate the entropy measure ‘p1’ and ‘p2’ for the two compared emotions if there is a heterogeneity condition.

```
INPUT: {DM_rms1, DM_rms2, DM_mean1, DM_mean2}
OUTPUT : {Emotion()}
BEGIN:
If ((DM_rms1<DM_rms2) AND (DM_mean1<DMs_mean2)) then
    Test data = > emotion A
Else if ((DM_rms1>DM_rms2) AND (DM_mean1>DM_mean2)) then
    Test data => emotion B
Else if (none of the above satisfies) then
 // here the homogeneity is not there with the  adjacent condition
  // hence calculating the Entropy Measure similar to the classical method
p1 = ((-1)*DM_rms1*(log (DM_rms1)/log (2)))-(DM_mean1*(log (DM_mean1)/log (2)));
p2 = ((-1)*DM_rms2*(log (DM_rms2)/log (2)))-(DM_mean2*(log (DM_mean2)/log (2)));
If (p1<p2) then                    // p1 and p2 are the two entropy measures
   Test data => emotion A
Else
   Test data => emotion B
End If
```

Table 4.3.5.2.1: CDT Homogeneity Algorithm

Hence by using the above algorithm, the emotion classes of the test dataset can be evaluated and the results of which are discussed in the later part. The algorithm looks very simple and less complex in structure and thus the question arises whether this transformation is suitable for this whole huge dataset. The question can be answered only by pure experimentation and testing by giving the known trained data as an input and make the model to predict the same.

## 4.4 Inducing Emotions to Modified POMDP

Unlike previous POMDP approaches our method provides the services considering all factors from the perspective of the user including his/her emotions. This is to achieve the optimal goal with few dialogue states in shorter time. We have also concentrated more on pruning the number of dialogue states at least by ten percent depending upon the type of conversation in the domain. Our dialogue manager has the same components exactly like the previous dialogue managers composed of such knowledge base, updated dialogue history, discourse generator and session model.

The system starts with the greet message followed by the system query to request for what type service to be provided to the user. Initially, at time'tʼ the system is normally in some unobserved state, $s \varepsilon S$. When the conversation established between the user and the system, the dialogue state takes a transition from s to s' by the increment of time stamps. Choosing the necessary action for the dialogue in our system is dominated by the four control modes in our system. The decision making or the switch between the control modes depends on the time available to make decisions on the particular context of the dialogue. Hence we have introduced a factor *TA* which represents the available time for choosing the essential action which depends on the machine state $s_m$, set of observations $o'$, machine actions $a_m$ and the belief state $b(s)$ of the machine, as denoted by $TA(S_m, a_m, o', b)$ at each time $t$. Depending upon the values of TA, and the machine states $s_m$ the switching between the modes takes place, which does not mean that the decision is taken now. The decision making is done by comparing the state of the system $s_m$, machine actions $a_m$, observation o' which is in combination of user action and user emotion {Ua + Ue}, belief state $b$ and the type of control mode TA at present the dialogue is in (i.e. at time *t).* To calculate the belief distribution of the dialogue, we have also introduced a factor $t_r$ which represents the response time of the system to the user in milliseconds. Hence the belief state distribution is updated based on $o'$ and action $a$ as follows:

$$\mathrm{B}'(\mathrm{S}') = \frac{\sum_{s\,\varepsilon\,S} \mathrm{b(s)T(s,a,s')O(a,s',z)t_r(s)}}{O(a,s',z)}$$

Here, the dialogue states and actions represent the machine states and machine actions respectively. And the value of TA depends on the current action and belief state distribution, which is given by TA(O'(u + e) | s, a, b'). Based on the current belief state and available time, the machine selects an action $a \ \varepsilon$ A, receives a reward r($s_m$, $a_m$) and transitions to a new unobserved state s'. Then, the system receives an observation o' $\varepsilon$ O depends on the system state $s_m$ and the system action $a_m$. Finally, the belief state is updated with a new one at particular time *t.*

### 4.4.1 Rewards

We have also changed the reward model depending upon the modified POMDP approach with four control modes. Previous POMDP model has two types of reward with some positive values for correct dialogue, i.e., the system exactly understand the user utterances and provides exactly the service as what the user wants. Negative values or zero stands for incorrect dialogue in which the system does not understand the user utterances exactly. In our approach, we have represented the rewards type depending upon the control modes: +100 if the dialogue state is in strategic mode and tactical mode as the system understands what the user wants, or -100 if the dialogue is in opportunistic mode as the user does not provide correct information or if there any conflict in the information provided by the user. Here, the system receives a negative reward and an instruction to provide options to the user. A zero reward means the system is in the scrambled mode because, in this mode, the system does not receive any proper information or query from the user but some disturbances, error or some corrupted information. λ is used as discount factor at time t, and the reward R is given by,

$$R = \sum_{t=0}^{\infty} \lambda^{t} r(b_t, a_m, t_r) \ \sum_{t=0}^{\infty} \lambda^{t} \sum_{s \, \varepsilon \, S} b_t(s) t_r(s) \, r(s, a_m, t)$$

Each action is determined by a policy π and POMDP system involves in finding the optimal policy $\pi$ * for the application which maximizes the rewards.

$$\pi^{*}(s_t) = a[argmax_{a \varepsilon A}(b_t, t_r)]$$

## 4.4.2 Confidence Score Evaluation

We have also incorporated the confidence score by providing an estimation of real value to show how exactly the system understands the user utterances as denoted by *c* with a predefined threshold value 0, which in turn affects the rewards received for each dialogue state. We have not made any changes to the confidence buckets as it depends on the user's utterances and system’s observation of user action $a_u$. We have used the same evaluation for calculating confidence score as it does not make any change in choosing system actions. But we have included an option of paraphrasing or double checking mechanism to increase the confidence scores. In this case, the system transitions to tactical mode because the system reconfirms the user utterance by providing an option in order to understand the user requirement exactly and reach the optimal goal within the available time. A graphical flowchart representation of our proposed work and its pseudo code are given as follows.

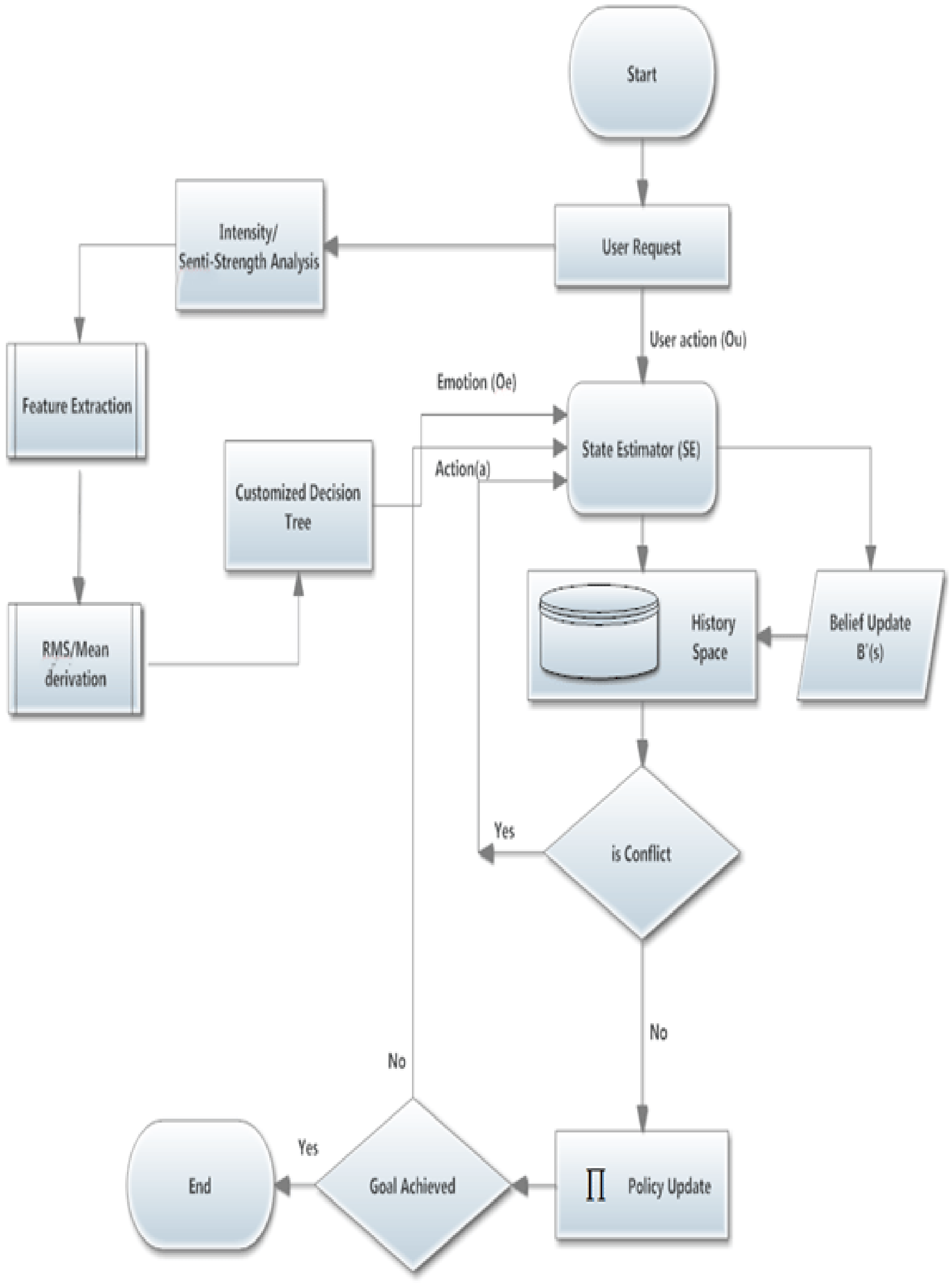


Figure 4.4.2.1: Flow Chart Representation of Proposed Work

```
EMPOMDP (bel(s), o_u, o_e, a):
INPUT: {Belief State (s), Observation list (User Action, User Emotion, User Goal),
action (a)}
OUTPUT: {s', Belief State'(s'), Scenario (S)}
CURRENT_BELIEF:= bel (s)
For all States (s) do
    Bel'(s') = $\alpha$ $P((o_u + o_e)_{t+1} \mid s_{t+1}, a_t)$ $s \, \mathcal{E} \, S \sum_{i=0}^{t}$ $P(s_{i+1} \mid s_i, a_i, O^e_i)$ $b_i(s)$
    If (Bel'(s') < 0.05) then
    {Ignore state of belief}
    Else {
        SCN: = Generate Scenario (Bel'(s), Bel(s))
        Domain_Constrain_validation (Bel'(s))
        If (! VALIDATION.FAIL) && (SCN.Equals (USER.GOAL)){
            Machine action = OUT.POLICY (Bel'(s))
            Return (s', Bel'(s'), SCN)
        }
        Else {
Machinenext_action= a with the hint
            Bel'(s) = CURRENT_BELIEF
            return Machin_next_action
        }
End For
```

Figure 4.4.2.2: Pseudo Code for New Approach

In the pseudo code of Figure 4.4.2.2, the *Bel(s),* Oe, Ou and *a* are the inputs of the proposed method. *Bel(s)* is the previous belief state of the last stage, o is the up-to-date observation and *a* is the last action taken by the machine. By recording the previous belief state, the belief state will be updated based on the POMDP theory and for all the belief states with possibility less than 0.05. For all the rest of the belief states, the domain constraint validation process function *DomainConstrain.validation()* will be invoked to check the conflicts of the belief states. The failure validation will result in the action to require further information with hints and the roll back of the belief states. Otherwise the action produced by the original POMDP solution policy *out.policy (Bel'(s))* will be taken.

## 4.5 Conclusion

In this chapter, we have discussed about the contributions made in the POMDP based dialogue management systems to make dynamic decision making depending on the control modes of the approach with respect to observation, emotions and current state of belief. We have also presented the modified approach of POMDP for handling real world state and improbability. As well, we have discussed how our approach extends the reward model and confidence scores. The main advantage of our proposed model is the robustness compared to the different dialogue management approaches.

# CHAPTER V

# EXPERIMENTATION AND RESULTS

This chapter will review the effectiveness and efficiency of the proposed system. The experimental analysis has been subdivided in two sections. The beginning of this chapter will discuss about the qualitative analysis in detail carried out between the proposed system and some of the benchmarks and the baselines. The order of further explanation here are as follows: The domain background applied in our case study will be introduced first and the implementation platform, utilized tools and corresponding details will be explained after. The results for the emotion recognition process are also explained to understand the efficient emotion induction to the POMDP model. Finally, the results under different possible scenarios, analysis of the outcome and comparison with the baselines will be given.

## 5.1 Implementation Setup for Emotion Recognition

In this section the detailed implementation and experimentation results of the classification  of emotions are discussed, analyzed and the resutls are shown. As per our earlier discussion there are four main implementation required for this anaylsis:

- ✓ Intensity Evaluation
- ✓ Senti-Strength Analysis
- ✓ Secondary Dataset Preparation
- ✓ Emotion Recognition using Customized Decision Tree (CDT)

### 5.1.1 Intensity Evaluation

Intensity of the emotion is calculated and updated in the dataset. This gives more information about the data for the machine to learn so that the efficiency of the learning is expected to increase. Two datasets from digg website [42] and SemEval Affective Text 2007 [43] are studied. The emotion texts are collected from both the datasets. Intensity of the emotion is analyzed with an interface as follows:

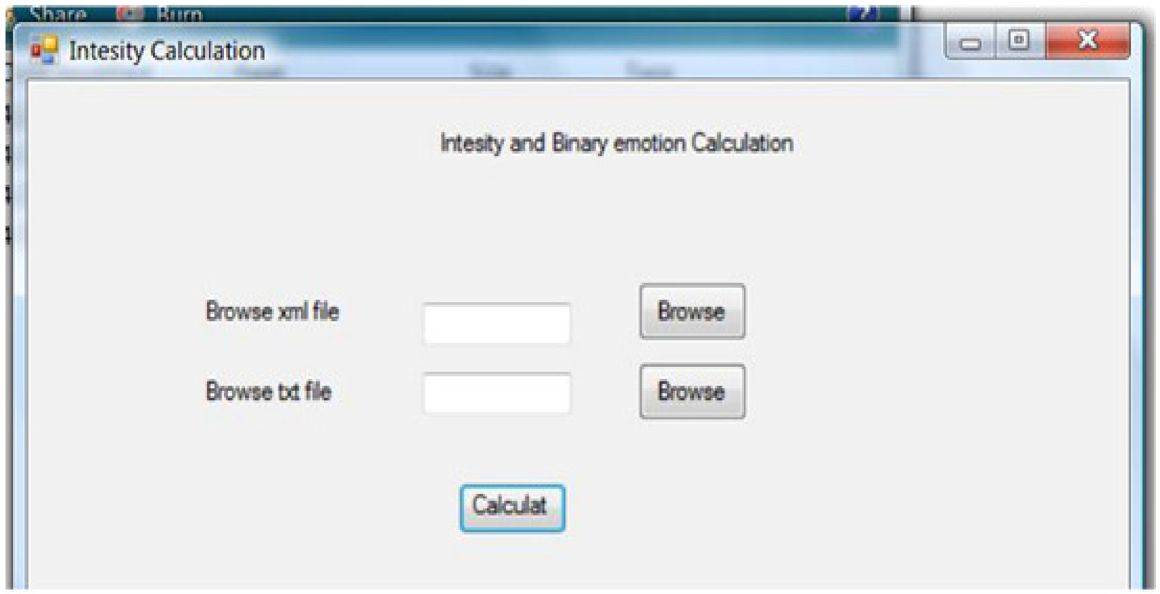


Figure 5.1.1.1: Intensity and Binary Emotion Calculation

The above is a C# application which calculates the values either from an XML/Text inputs as it has to take care of Textual or Speech grid inputs. The intensity of texts for example deals with keywords, emoticons, boldness of the letters and cases of the text sense. These values are scaled initially from 1 to 5 which sense lower to a higher emotional intensity as shown in Figure 5.1.1.2. It does not necessarily have to have fixed intervals for six primary emotions as intensity can be lower to higher levels for each of them individually. When the intensity value is higher, the prediction of emotion becomes better. The results for this evaluation are given after discussing the other parts.

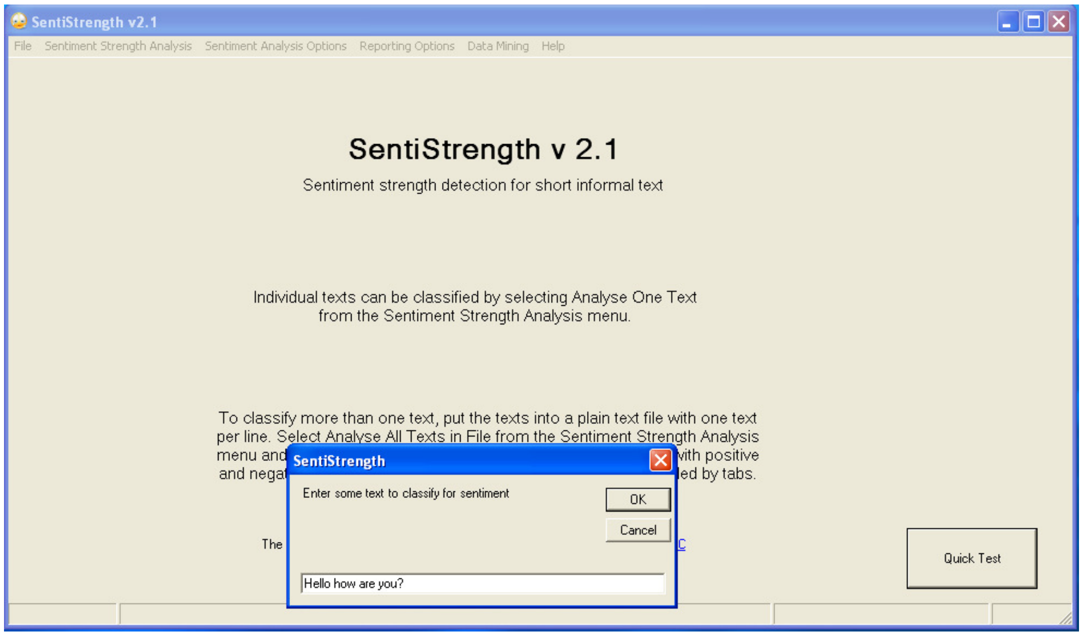
| Data 1 | Data 2 | Data 3 | Data 4 |
|---|---|---|---|
| 2.028 | 2.374 | 2.06 | 2.371 |
| 2.05 | 2.349 | 2.077 | 2.422 |
| 2.029 | 2.336 | 2.077 | 2.422 |
| 2.03 | 2.338 | 2.082 | 2.383 |
| 2.036 | 2.346 | 2.059 | 2.399 |
| 2.078 | 2.388 | 2.091 | 2.402 |
| 2.036 | 2.328 | 2.098 | 2.409 |

Table 5.1.1.2: Intensity Values Metric (Partial) [1]

### 5.1.2 Senti-Strength Analysis

Reiterating the earlier statement, positive and negative emotion strengths are updated to the database by the senti-strength software analyzer. For each text, the senti-strength output is two integers: 1 to 5 for positive sentiment strength and a split score of -1 to -5 for negative sentiment strength. Here, 1 or -1 signifies least sentiment and 5 or -5 signify strong sentiment of each type. For example, a text with a score of [3, -5] would include moderate positive sentiment and strong negative sentiment. A neutral text would be inferred as [1, - 1]. Two scales are used because even short texts can enclose both positivity and negativity. The objective is to perceive the sentiment expressed rather than its overall polarity [42].

Figure 5.1.2.2 and Figure 5.1.2.3 show the implementation setup of the senti-strength analysis.


SentiStrength v2.1
File Sentiment Strength Analysis Sentiment Analysis Options Reporting Options Data Mining Help
SentiStrength
OK
Cancel
Quick Test

SentiStrength v 2.1

Sentiment strength detection for short informal text

Individual texts can be classified by selecting Analyse One Text from the Sentiment Strength Analysis menu.

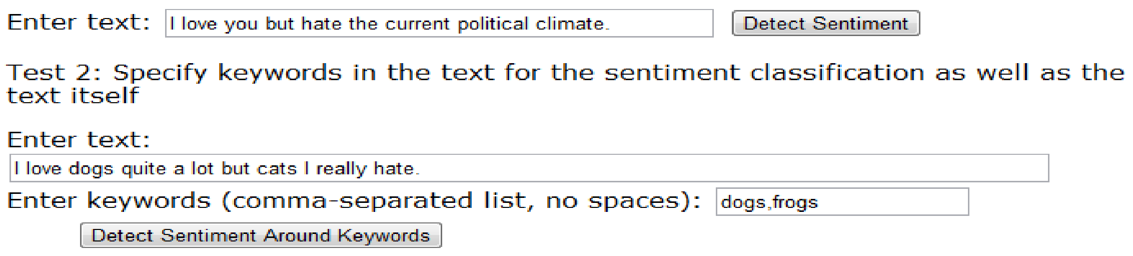
To classify more than one text, put the texts into a plain text file with one text per line. Select Analyse All Texts in File from the Sentiment Strength Analysis menu and ... with positive and negat... ed by tabs.

The

Enter some text to classify for sentiment

Hello how are you?

Figure 5.1.2.1: Text Input to Senti-Strength [42]

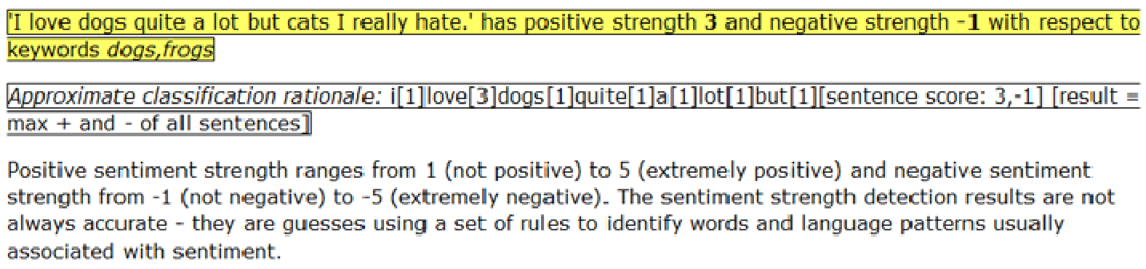


Enter text: I love you but hate the current political climate.

Test 2: Specify keywords in the text for the sentiment classification as well as the text itself

Enter text:
I love dogs quite a lot but cats I really hate.
Enter keywords (comma-separated list, no spaces): dogs,frogs

Figure 5.1.2.2: Text Input to Senti-Strength



'I love dogs quite a lot but cats I really hate.' has positive strength **3** and negative strength **-1** with respect to keywords *dogs,frogs*

*Approximate classification rationale:* i[1]love[3]dogs[1]quite[1]a[1]lot[1]but[1][sentence score: 3,-1] [result = max + and - of all sentences]

Positive sentiment strength ranges from 1 (not positive) to 5 (extremely positive) and negative sentiment strength from -1 (not negative) to -5 (extremely negative). The sentiment strength detection results are not always accurate - they are guesses using a set of rules to identify words and language patterns usually associated with sentiment.

Figure 5.1.2.3: Positive and Negative Strength Output

### 5.1.3 Secondary Emotional Dataset

The datasets used in this project have seven columns with different emotional strengths for a sentence and the intensity value. The emotional strengths are such as strength of angry, strength of surprise, strength of sadness, strength of fear, strength of disgust, strength of joy. These seven columns make the decision for a particular statement's emotion. They are as well the key elements or the attributes in determining textual emotions and hence can be called as features hereafter. Strength of emotions in the training set is done by human annotators. The primary datasets are obtained from two sources: SemEval-2007 Task 14: Affective Text [43] and Cyber emotions [42]. The secondary dataset for analysis is obtained with the combination of the features used in the emotion classification dataset and intensity as follows:

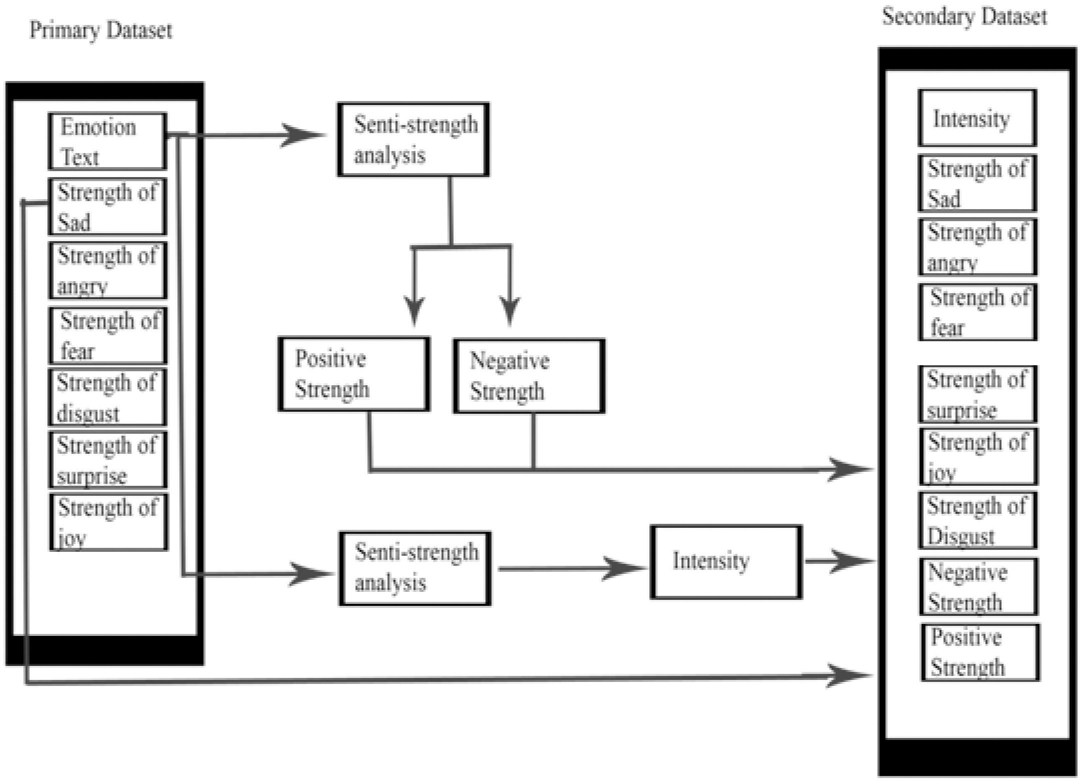


Figure 5.1.3.1: Secondary Dataset Generation

The secondary dataset is used for experimentation of the text emotion analysis. The emotional intensity calculation and experimentation are done using C# application. Datasets are structured with necessary features using Kettle software [44] as shown in Figure 5.1.3.2.

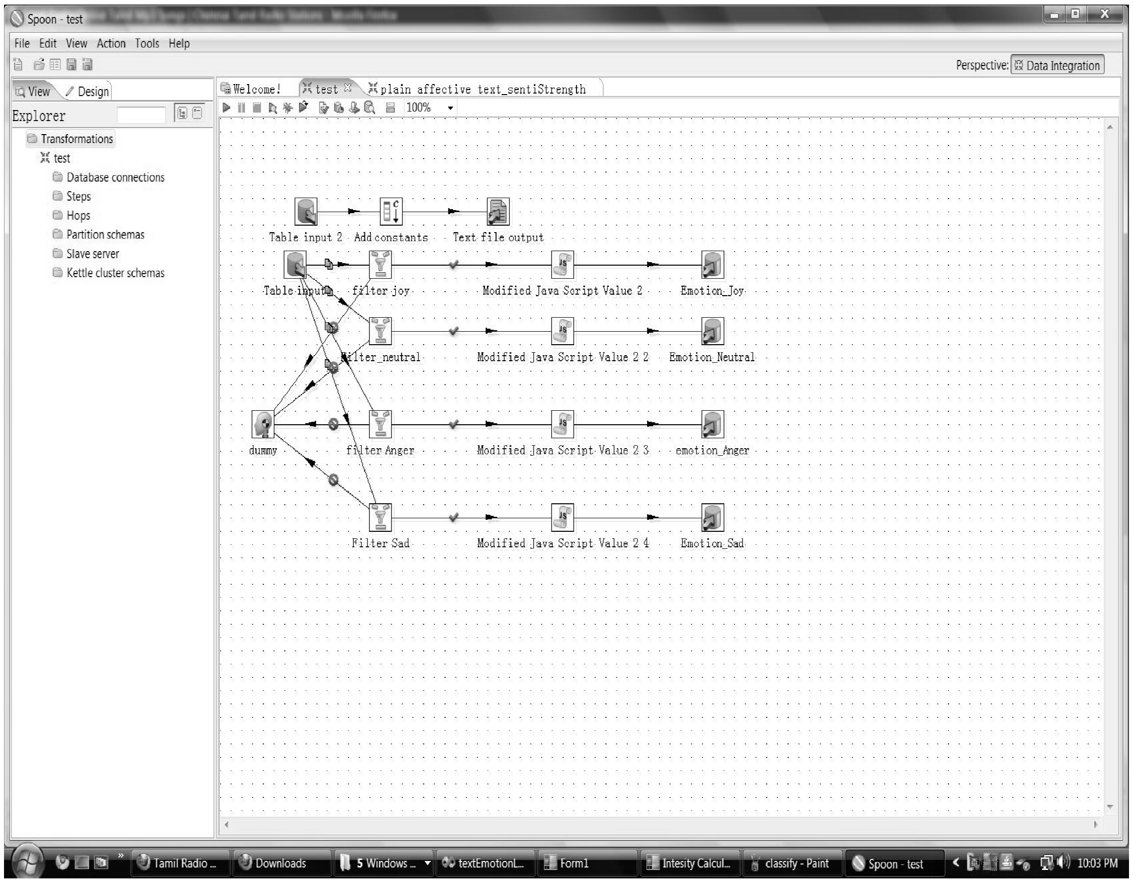


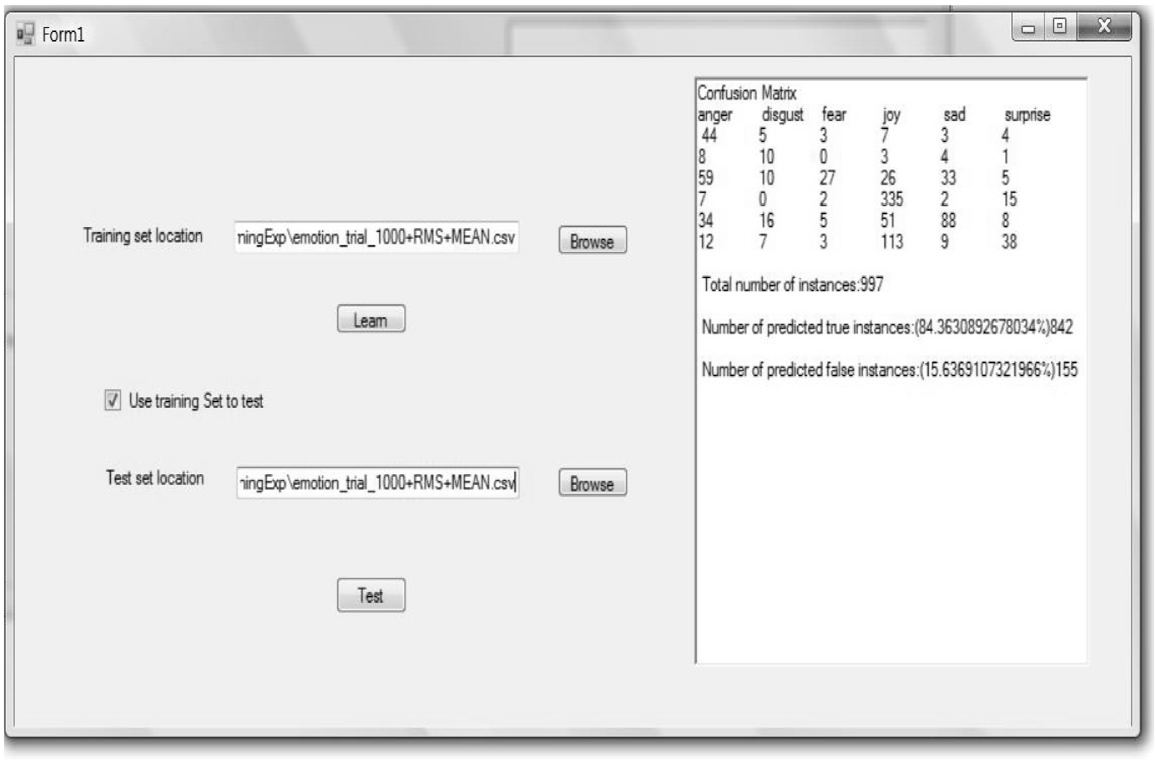
Figure 5.1.3.2: Extraction of Data from two Different Sources

## 5.1.4 Customized Decision Tree Implementation

The implementation of the customized decision tree classifier has to evaluate the overall mean and RMS values of seven features namely: Anger, Disgust, Fear, Joy, Sad, Surprise and Intensity. This is done in Visual studio C# .Net and the graphical representation using the Microsoft Excel for comparison. After calculating the mean and the RMS values, the graphs were plotted between the seven classes to identify the difference maintained between each of those classes. There are 4 modules implemented for this experiment as follows:

1. Mean-RMS dataset generation
2. Finding the set of possible emotions.
3. Updating the emotion points.
4. Finalizing the emotion class.

The secondary dataset is derived from multiple datasets using the Senti-analysis and intensity calculation as described earlier in this thesis. This secondary dataset is used as the input to the C# implemented windows application as shown in Figure 5.1.4.1.



Training set location: ningExp\emotion_trial_1000+RMS+MEAN.csv

[✓] Use training Set to test

Test set location: ningExp\emotion_trial_1000+RMS+MEAN.csv

```
Confusion Matrix
anger    disgust   fear    joy    sad    surprise
 44      5         3       7      3      4
8        10        0       3      4      1
59       10        27      26     33     5
7        0         2       335    2      15
34       16        5       51     88     8
12       7         3       113    9      38

 Total number of instances:997

 Number of predicted true instances:(84.3630892678034%)842

 Number of predicted false instances:(15.6369107321966%)155
```

Figure 5.1.4.1: C# Application for Experimenting CDT

The implemented application uses an intermediate dataset to calculate the mean and the RMS for every feature in the secondary dataset. Let this intermediate dataset be referred as Mean-RMS Dataset which is used for the classification of emotion. Thus, after running the customized decision tree algorithm for evaluating the overall mean and the RMS values, the emotions are predicted in the mixed dataset. The Mean-RMS dataset is with 7 rows of mean and RMS. After learning the Mean-RMS dataset, the machine could then be able to classify the emotions from the test dataset. Test data is browsed with the windows application developed and then the test option is selected. This creates a new file with the predicted emotions in the last column of the Mean-RMS dataset.

### 5.1.5 Customized Decision Tree Evaluation

The graphs below show the variation of RMS values of the emotion class dimensions. The dimensions are plotted along the 'x' axis as follows: strength of angry, strength of surprise, strength of sadness, strength of fear, strength of disgust, strength of joy and intensity of emotion. The mean, the RMS and respective random emotion data are plotted on the 'y' axis. The graphical pictures demonstrate that the difference between the two classes lies only in certain dimension values, which indicates that the classification process needs a better classifier to make the exact prediction. Hence it is evident that using any classification technique, it is practically impossible to provide 100% accuracy in predicting the emotional data.

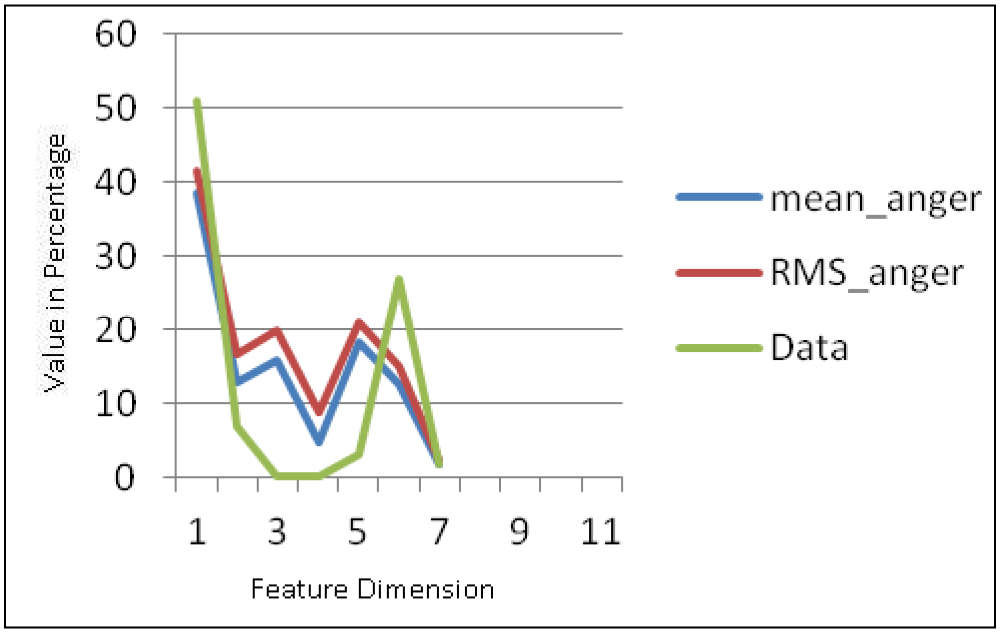


Figure 5.1.5.1: Anger Emotion Class with Mean and RMS

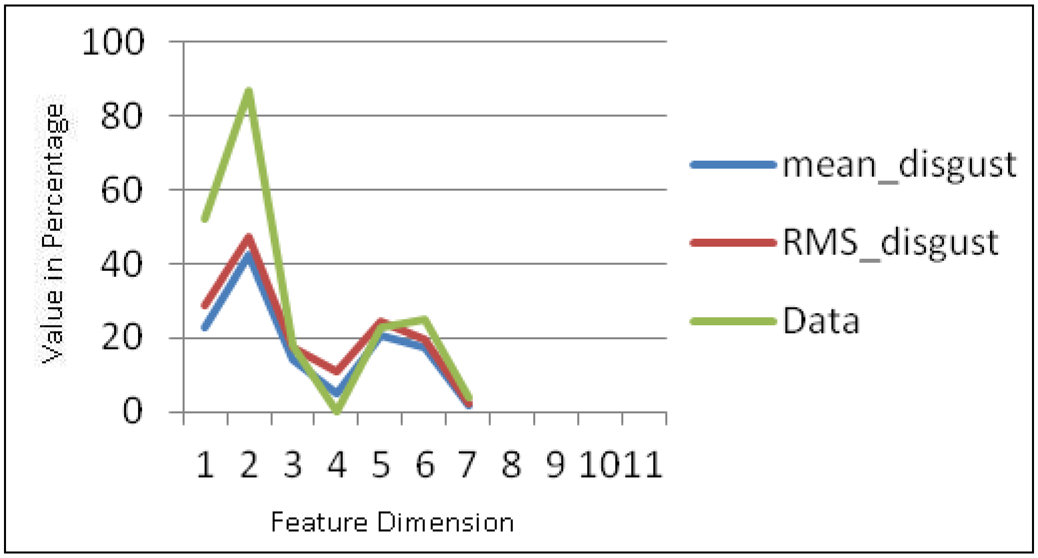


Figure 5.1.5.2: Disgust Emotion Class with Mean and RMS

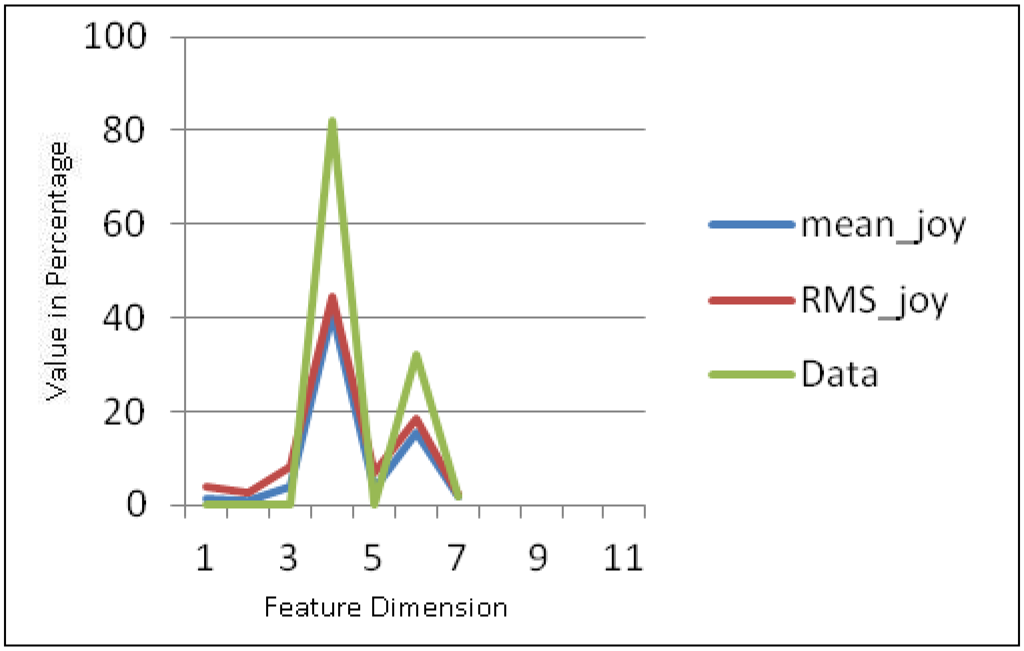


Figure 5.1.5.3: Joy Emotion Class with Mean and RMS

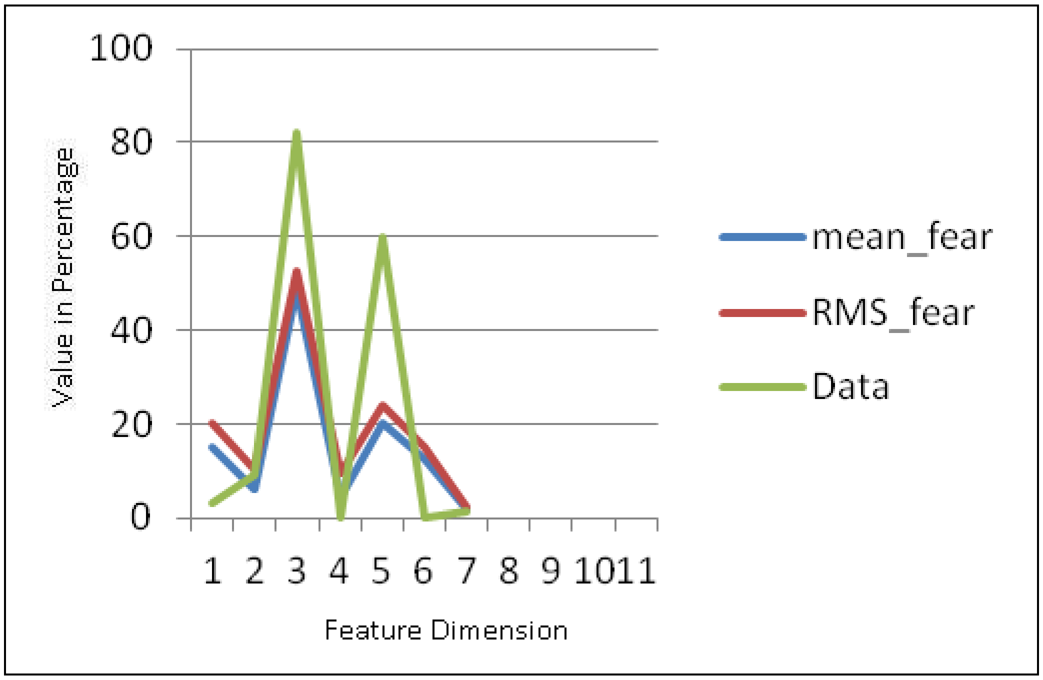


Figure 5.1.5.4: Fear Emotion Class with Mean and RMS

Consequently, running the customized decision tree algorithm after calculating the mean and the RMS values to predict the emotion class in the mixed dataset, it predicted 926/1005 in joy class, 630/996 surprise class, 661/928 sad class and 884/1042 anger. Fear class and disgust class have very small numbers in training so they cannot be taken into account for calculating the efficiency of the classifier. The above graphical results indicate that the classification works better with respect to the knowledge gained from the datasets. The prediction process could be improved by training the system with large feeds of known data or improving the classifier with further modifications, which are discussed in Section 5.3.

| **Mixed Data** | **Emotion** | **Occurrence** | **Homogeneity count** | **Entropy measures** |
|---|---|---|---|---|
| Data1 | Joy | 5 | 0 | |
| Data2 | Disgust | 4 | 1 | p1=-185.8704 p2=-185.2176 |
| Data3 | Joy | 6 | 1 | |
| Data9 | Joy | 8 | 2 | |
| Data10 | Anger | 4 | 0 | |
| Data11 | Sad | 3 | 0 | |
| Data12 | Surprise | 7 | 1 | |
| Data13 | Fear | 11 | 0 | |

Table 5.1.5.5: Predicted Unknown Emotion (CDT)

### 5.1.6 Validation of Customized Decision Tree

The simple and easy way to validate the algorithm is to induce the known emotion dataset into the prediction model and calculate the accuracy using it. We can calculate 'True Positive' and 'False Positive' value and then build a matrix between the True positive ratio (TPR) and the random data. The machine is trained with close to 1000 data in each of the emotion set, and a dataset is prepared whose emotions are known and given for prediction. The final predicted emotion for every data is noted and validated with the originally obtained emotion. The implemented C# application has functionality of testing the algorithm with unknown emotion. The predicted emotions are verified with the emotion classified by an annotator. The validated result is represented in the form of confusion matrix with a small mixed set example of close to 900 mixed set of emotions.

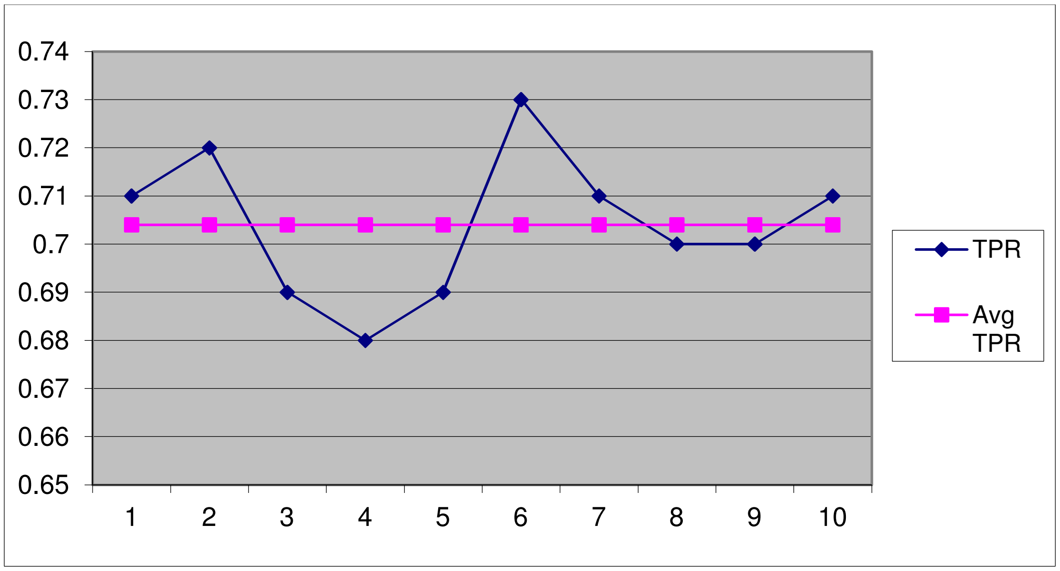


Figure 5.1.6.1: True Positive Ratio Graph for 10 Subsets of Training Sets

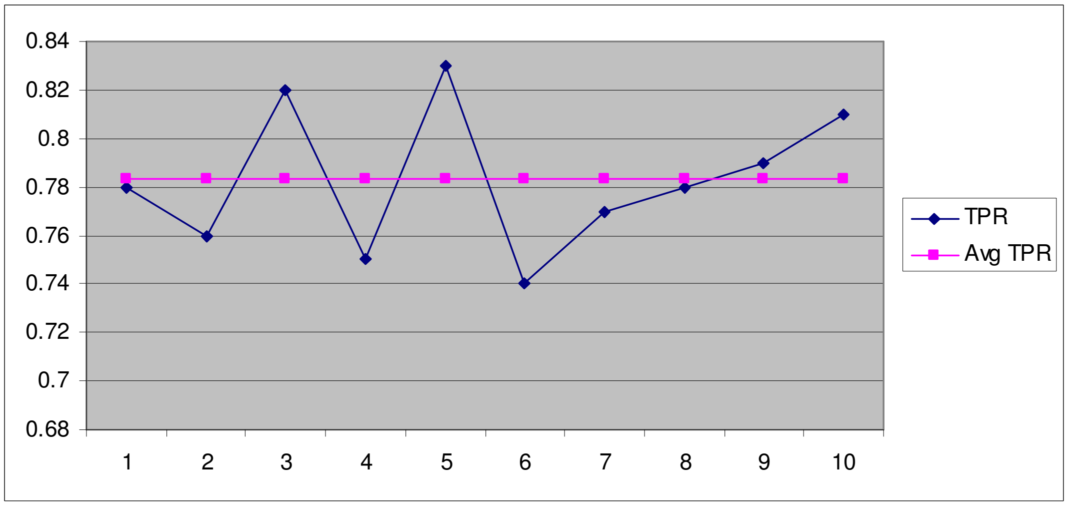


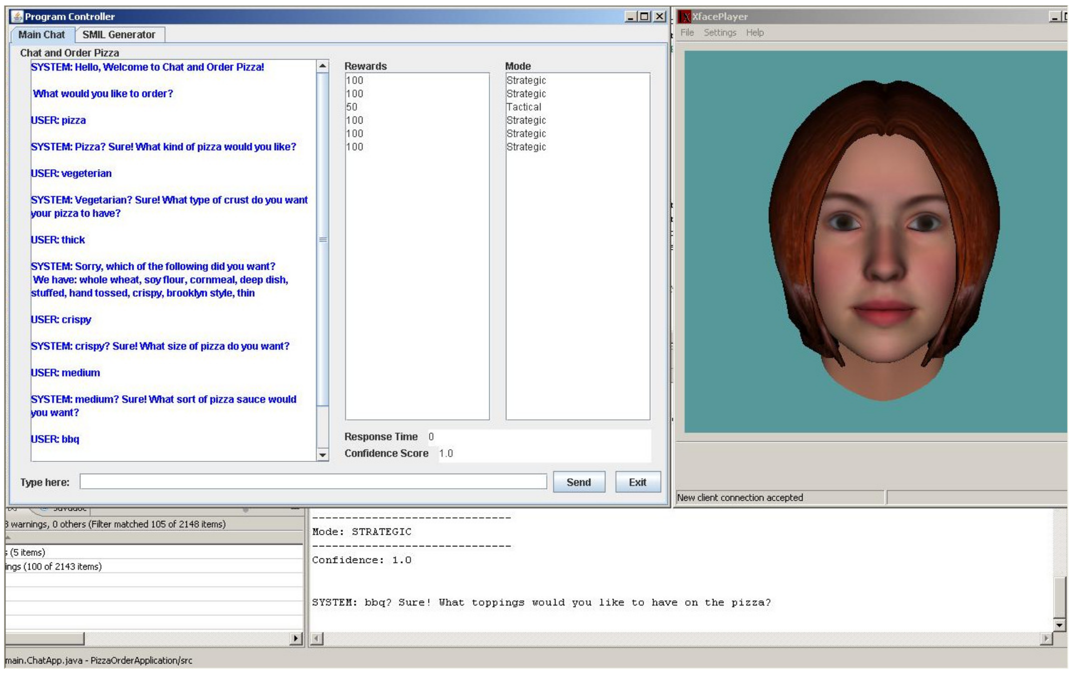
Figure 5.1.6.2: True Positive Ratio Graph for 10 Subsets of Test Sets

|  | anger | disgust | fear | joy | sad | surprise | Percentage |
|---|---|---|---|---|---|---|---|
| anger | 48 | 5 | 3 | 7 | 3 | 4 | 72% |
| disgust | 8 | 18 | 0 | 0 | 4 | 1 | 59% |
| fear | 27 | 10 | 108 | 26 | 33 | 5 | 61% |
| joy | 7 | 0 | 2 | 335 | 2 | 15 | 92% |
| sad | 34 | 16 | 5 | 21 | 156 | 8 | 65% |
| surprise | 12 | 7 | 3 | 38 | 9 | 131 | 65% |

Table 5.1.6.3: Confusion Matrix on Test Set

### 5.1.7 Quantitative Analysis

The purpose of this pizza ordering system will be to provide a client service by considering the customer requirements and intentions. The creation of such a system with a 3D talking head driven by emotions will make it more engaging to the end user. This motivated us to create a system that combines the current state-of-the-art 3D facial animation with a spoken dialogue system, along with a new cognitive model for generating emotion of the user trying to interact and order the pizza. Thus the new algorithm was incorporated to find the user emotion after each transaction was completed. We achieved significant results to find the human emotions after ordering the pizza using this application.



Chat and Order Pizza

```
SYSTEM: Hello, Welcome to Chat and Order Pizza!

 What would you like to order?

USER: pizza

SYSTEM: Pizza? Sure! What kind of pizza would you like?

USER: vegeterian

SYSTEM: Vegetarian? Sure! What type of crust do you want
your pizza to have?

USER: thick

SYSTEM: Sorry, which of the following did you want?
 We have: whole wheat, soy flour, cornmeal, deep dish,
stuffed, hand tossed, crispy, brooklyn style, thin

USER: crispy

SYSTEM: crispy? Sure! What size of pizza do you want?

USER: medium

SYSTEM: medium? Sure! What sort of pizza sauce would
you want?

USER: bbq
```

| Rewards | Mode |
|---|---|
| 100 | Strategic |
| 100 | Strategic |
| 50 | Tactical |
| 100 | Strategic |
| 100 | Strategic |
| 100 | Strategic |

Response Time 0
Confidence Score 1.0

New client connection accepted

```
3 warnings, 0 others (Filter matched 105 of 2148 items)
: (5 items)
ings (100 of 2143 items)
```

```
------------------------------
Mode: STRATEGIC
------------------------------
Confidence: 1.0


SYSTEM: bbq? Sure! What toppings would you like to have on the pizza?
```

main.ChatApp.java - PizzaOrderApplication/src

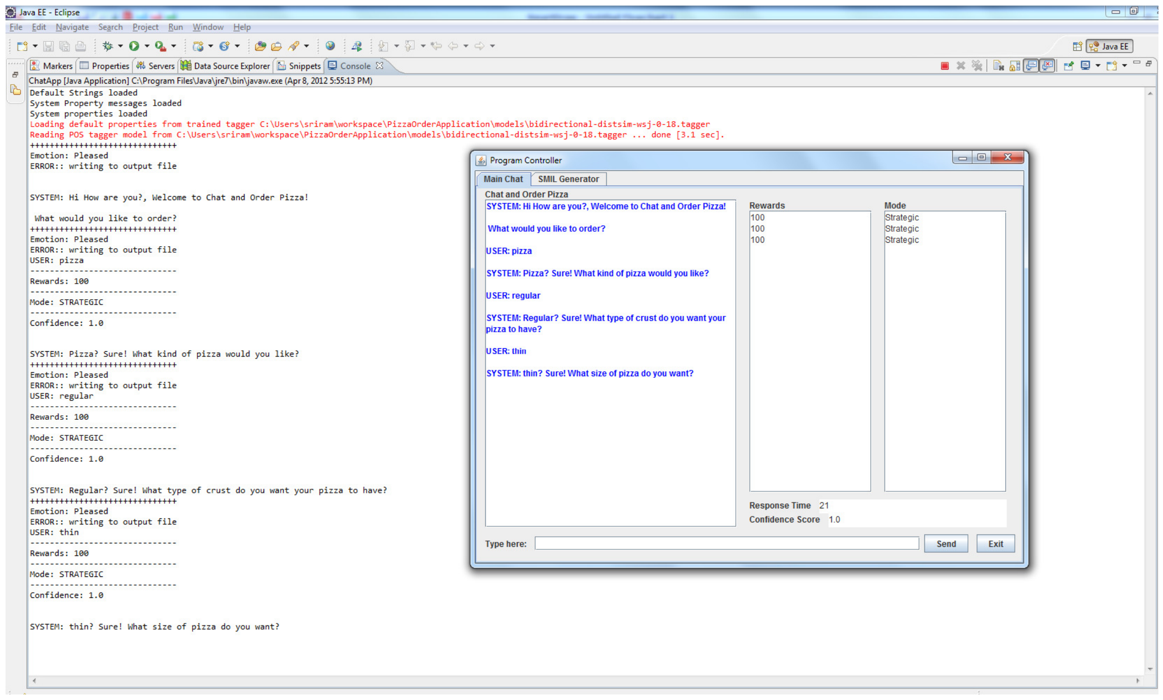
Figure 5.1.7.1: Pizza Order Application for Experimentation

We ran several possible test cases and ran the algorithm for the user inputs, which were able to deduce the emotions joy, anger and sadness, pretty easily with the sentences used by the user. Most of the test cases came to neutral emotion (no-emotion) as the user typed in only one sentence like “pizza” for the question “what would you like to order?” and “large” for the question “what size of a pizza would like?” The algorithm could construct more accurate emotion of the user, if there is enough information provided. For example, “I would like to order a pizza please” instead of just “pizza”, in which case the overall emotion can be found more accurately by grading emotional scores and finding sentiment strengths of each conversation. The final results are published with the POMDP’s results which would essentially make more sense.

## 5.2 Implementation of Modified POMDP with Emotions

Experiments are conducted based upon a simulated situation in which an agent provides assistance at a pizza restaurant to a human user for the purchase of a pizza. Hence, before we explain how we generate test samples for the Modified POMDP approach we would like to explain in detail about the user simulator. This user simulator system is implemented using JAVA under Eclipse Indigo and the knowledge base has been designed using MY SQL, In addition, a connection has been established between both front end and back end applications and data transfer (ETL) is done using Pentaho Data Integration Kettle [44].

We could easily call the user simulator as a POMDP mock up as it imitates all the necessary steps that a POMDP has to do, in order to create an absolute test set to test the new system and compare it with the benchmark.


Java EE - Eclipse
File Edit Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console
Java EE
Program Controller
Main Chat SMIL Generator
Send
Exit

```
ChatApp [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Apr 8, 2012 5:55:13 PM)
Default Strings loaded
System Property messages loaded
System properties loaded
Loading default properties from trained tagger C:\Users\sriram\workspace\PizzaOrderApplication\models\bidirectional-distsim-wsj-0-18.tagger
Reading POS tagger model from C:\Users\sriram\workspace\PizzaOrderApplication\models\bidirectional-distsim-wsj-0-18.tagger ... done [3.1 sec].
+++++++++++++++++++++++++++++
Emotion: Pleased
ERROR:: writing to output file


SYSTEM: Hi How are you?, Welcome to Chat and Order Pizza!

 What would you like to order?
+++++++++++++++++++++++++++++
Emotion: Pleased
ERROR:: writing to output file
USER: pizza
------------------------------
Rewards: 100
------------------------------
Mode: STRATEGIC
------------------------------
Confidence: 1.0


SYSTEM: Pizza? Sure! What kind of pizza would you like?
+++++++++++++++++++++++++++++
Emotion: Pleased
ERROR:: writing to output file
USER: regular
------------------------------
Rewards: 100
------------------------------
Mode: STRATEGIC
------------------------------
Confidence: 1.0


SYSTEM: Regular? Sure! What type of crust do you want your pizza to have?
+++++++++++++++++++++++++++++
Emotion: Pleased
ERROR:: writing to output file
USER: thin
------------------------------
Rewards: 100
------------------------------
Mode: STRATEGIC
------------------------------
Confidence: 1.0


SYSTEM: thin? Sure! What size of pizza do you want?
```

Chat and Order Pizza

SYSTEM: Hi How are you?, Welcome to Chat and Order Pizza!

What would you like to order?

USER: pizza

SYSTEM: Pizza? Sure! What kind of pizza would you like?

USER: regular

SYSTEM: Regular? Sure! What type of crust do you want your pizza to have?

USER: thin

SYSTEM: thin? Sure! What size of pizza do you want?

| Rewards | Mode |
|---|---|
| 100 | Strategic |
| 100 | Strategic |
| 100 | Strategic |

Response Time 21
Confidence Score 1.0

Type here:

Figure 5.2.1: User Interface of User Simulator (POMDP Mock up)

In the user interface/chat screen, we have also displayed the emotion, rewards, mode of the dialogue and confidence scores just for our own tracking purpose to test whether the system performs efficiently. As we use the history space of the dialogue to make decisions or choose system actions, the developed dialogue mock up tracks the system actions, confidence scores, rewards, mode of the dialogue and transition between the modes, response time of the system for each of the dialogues and the belief states.

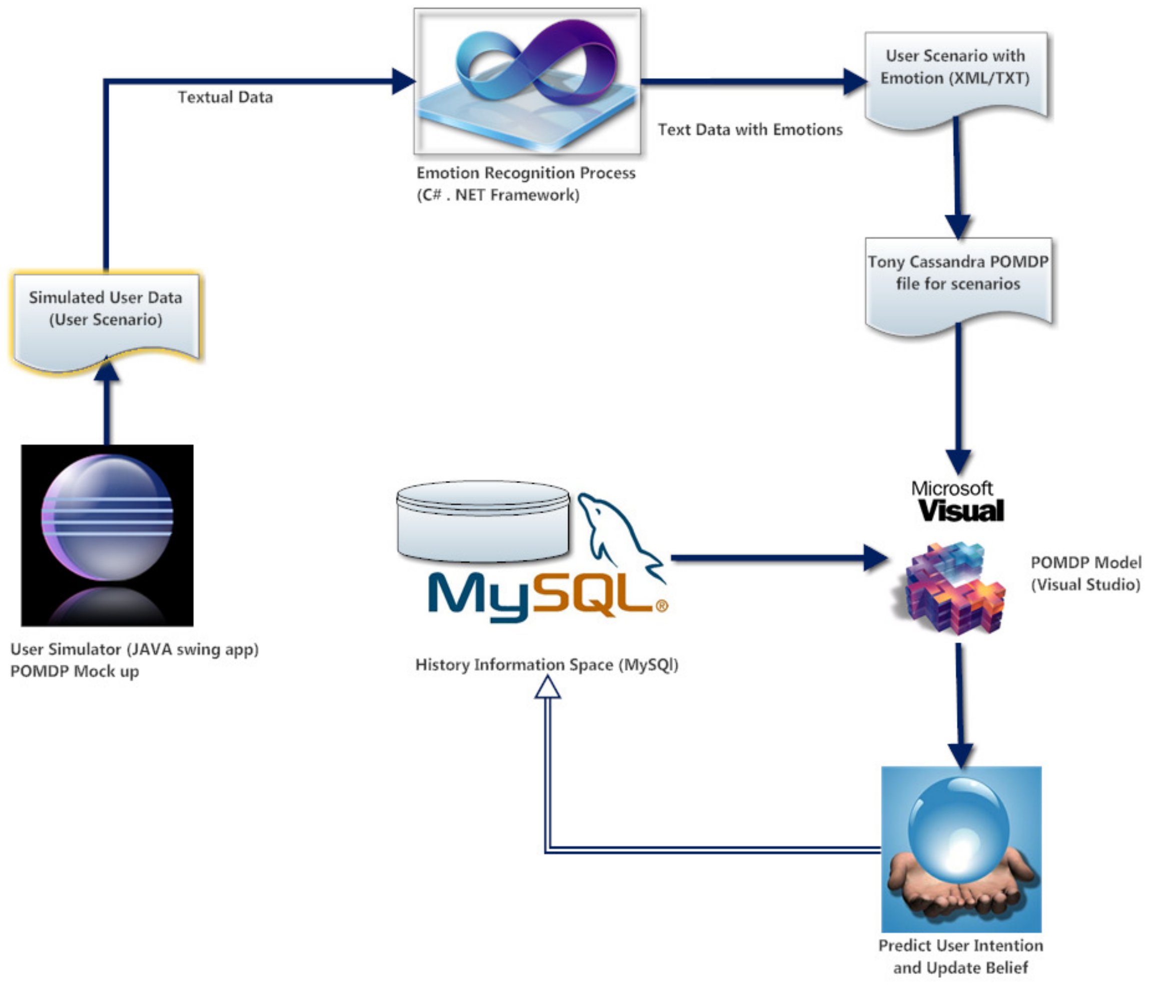
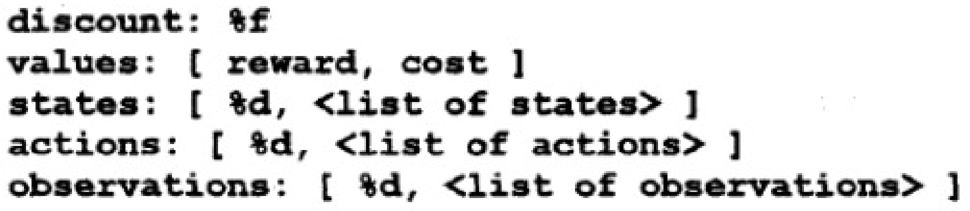

Textual Data
Emotion Recognition Process (C# . NET Framework)
Text Data with Emotions
User Scenario with Emotion (XML/TXT)
Tony Cassandra POMDP file for scenarios
Simulated User Data (User Scenario)
User Simulator (JAVA swing app) POMDP Mock up
MySQL
History Information Space (MySQl)
Microsoft Visual
POMDP Model (Visual Studio)
Predict User Intention and Update Belief

Figure 5.2.2: Overall Implementation Process of the Proposed Approach

## 5.2.1 POMDP File Specification

In the process of execution, the POMDP problem specification file is in the organization of Tony Cassandra [45] and the dialogue specification parser was developed by Bui [46] at the Human Media Interaction research group of the University of Twente is used. The POMDP's input file follows the Tony Cassandra's format [6] which can be handled by the POMDP solver. It is the formal problem specification file which encoded the domain problem under the distinct composition and semantics. Tony Cassandra POMDP specification file must have 5 important objects which specify the discount value, states, actions and observations at the beginning. Figure 5.2.1.1 shows the beginning objects definition. The order can be in any sequences and all of them must precede specifications of transition probabilities, observation probabilities and rewards.



```
discount: %f
values: [ reward, cost ]
states: [ %d, <list of states> ]
actions: [ %d, <list of actions> ]
observations: [ %d, <list of observations> ]
```

Figure 5.2.1.1: Tony Cassandra [41] File Objects

The transition possibilities can be specified in the following format:

*T: <action> : <start-state> : <end-state> %f*

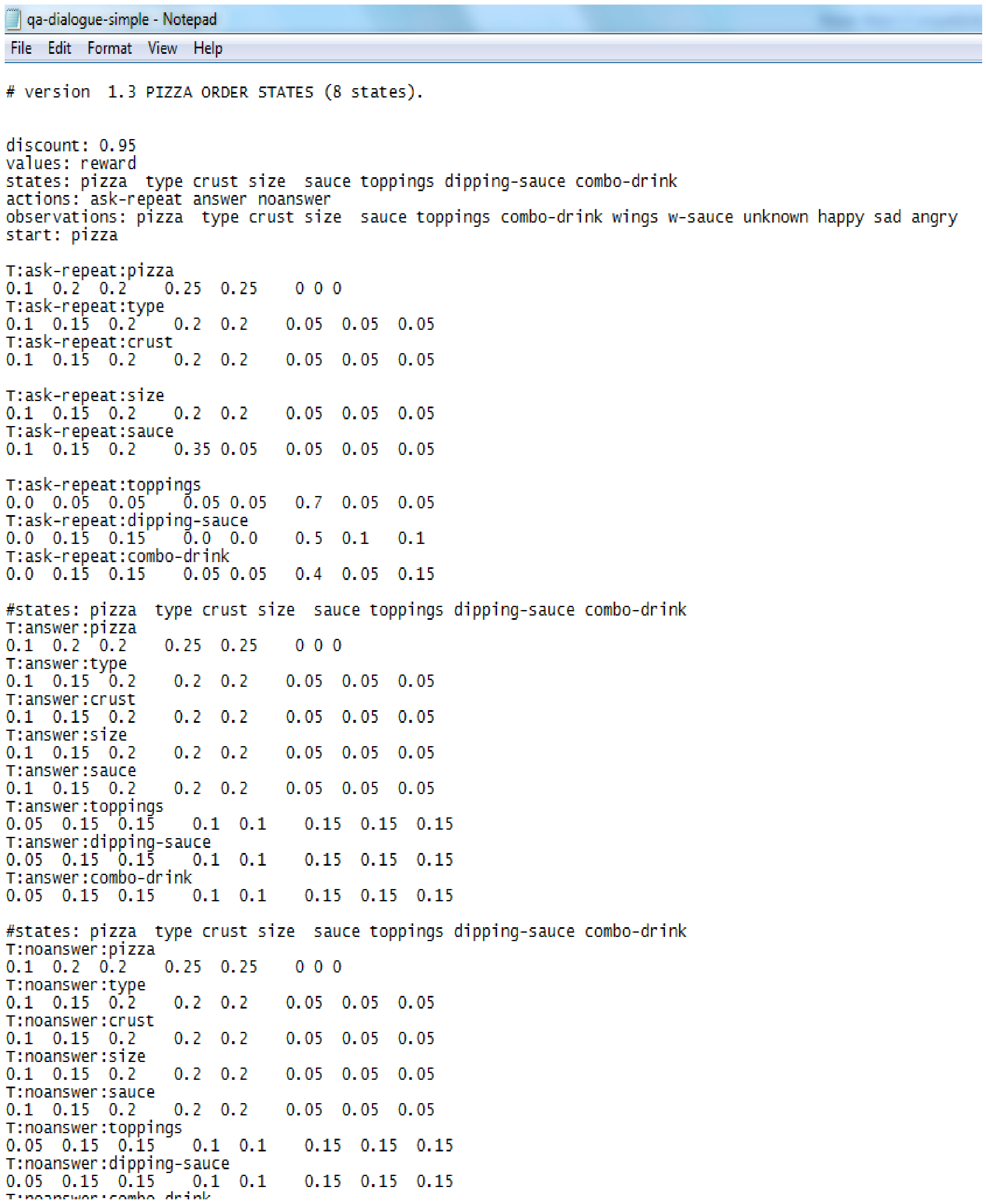
and observation probabilities are specified in a little similar way with transition probabilities in following format:

*O: <action>: <end-state> : <observation> %f*

The reward model is specified in this following format:

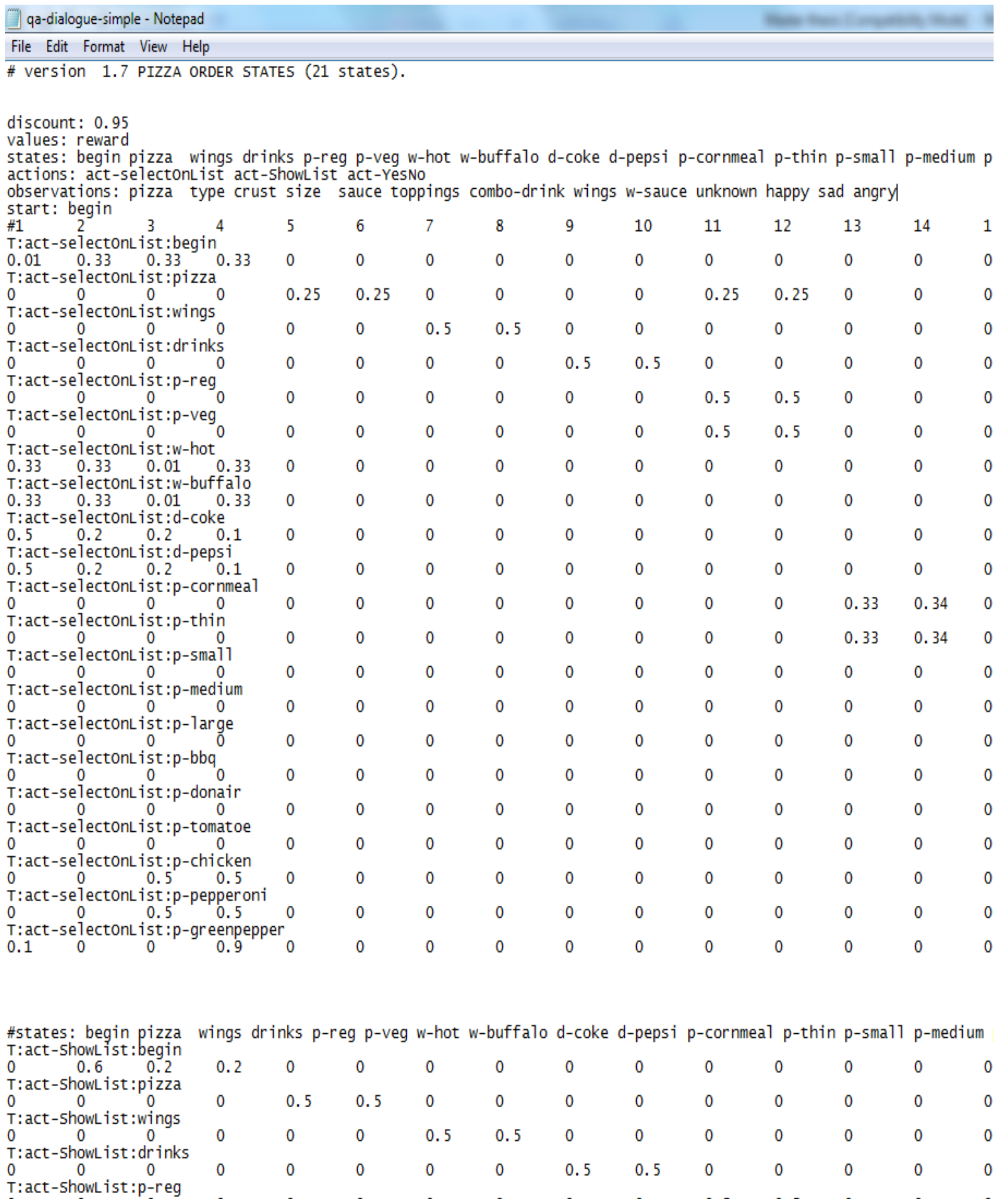
R: <action>: <start-state> : <end-state> : <observation> %f

For any of the entries appeared in the above, an asterisk * for either < state >, < action >, < observation > indicates a wildcard which means this item will be expanded to all existing entities. For the simulated pizza ordering system, the POMDP specification format is designed based on the experiences and domain knowledge. Since it is an individual POMDP file we specify a unique file for each of the user goals and user actions. The system can perform 4 types of actions. The number of the dialogue states are 21 including the begin state. The discount value is 0.95 in this experiment. The POMDP solver adopted in this experiment is ZMDP [47] solver. ZMDP is a software package which implements several heuristic search algorithms for POMDPs and MDPs developed by Trey Smith at the Carnegie Mellon University. ZMDP POMDP solver can work under both Linux and Mac operating systems. To solve the POMDP problem in our experiments, heuristic search value interaction algorithm (SARSOP) [48] is used. SAROPS is a point-based approximation algorithm that maintains both upper and lower bounds on the optimal value function, allowing it to use effective heuristics for action and observation selection, and to provide the policy that it generates. The following figures shows examples for POMDP file specification for pizza ordering system.



```
# version  1.3 PIZZA ORDER STATES (8 states).


discount: 0.95
values: reward
states: pizza  type crust size  sauce toppings dipping-sauce combo-drink
actions: ask-repeat answer noanswer
observations: pizza  type crust size  sauce toppings combo-drink wings w-sauce unknown happy sad angry
start: pizza

T:ask-repeat:pizza
0.1  0.2  0.2    0.25  0.25    0 0 0
T:ask-repeat:type
0.1  0.15  0.2    0.2  0.2    0.05  0.05  0.05
T:ask-repeat:crust
0.1  0.15  0.2    0.2  0.2    0.05  0.05  0.05

T:ask-repeat:size
0.1  0.15  0.2    0.2  0.2    0.05  0.05  0.05
T:ask-repeat:sauce
0.1  0.15  0.2    0.35 0.05   0.05  0.05  0.05

T:ask-repeat:toppings
0.0  0.05  0.05    0.05 0.05   0.7  0.05  0.05
T:ask-repeat:dipping-sauce
0.0  0.15  0.15    0.0  0.0    0.5  0.1   0.1
T:ask-repeat:combo-drink
0.0  0.15  0.15    0.05 0.05   0.4  0.05  0.15

#states: pizza  type crust size  sauce toppings dipping-sauce combo-drink
T:answer:pizza
0.1  0.2  0.2    0.25  0.25    0 0 0
T:answer:type
0.1  0.15  0.2    0.2  0.2    0.05  0.05  0.05
T:answer:crust
0.1  0.15  0.2    0.2  0.2    0.05  0.05  0.05
T:answer:size
0.1  0.15  0.2    0.2  0.2    0.05  0.05  0.05
T:answer:sauce
0.1  0.15  0.2    0.2  0.2    0.05  0.05  0.05
T:answer:toppings
0.05  0.15  0.15    0.1  0.1    0.15  0.15  0.15
T:answer:dipping-sauce
0.05  0.15  0.15    0.1  0.1    0.15  0.15  0.15
T:answer:combo-drink
0.05  0.15  0.15    0.1  0.1    0.15  0.15  0.15

#states: pizza  type crust size  sauce toppings dipping-sauce combo-drink
T:noanswer:pizza
0.1  0.2  0.2    0.25  0.25    0 0 0
T:noanswer:type
0.1  0.15  0.2    0.2  0.2    0.05  0.05  0.05
T:noanswer:crust
0.1  0.15  0.2    0.2  0.2    0.05  0.05  0.05
T:noanswer:size
0.1  0.15  0.2    0.2  0.2    0.05  0.05  0.05
T:noanswer:sauce
0.1  0.15  0.2    0.2  0.2    0.05  0.05  0.05
T:noanswer:toppings
0.05  0.15  0.15    0.1  0.1    0.15  0.15  0.15
T:noanswer:dipping-sauce
0.05  0.15  0.15    0.1  0.1    0.15  0.15  0.15
T:noanswer:combo-drink
```

Figure 5.2.1.2: Tony Cassandra File Objects with 8 States



qa-dialogue-simple - Notepad

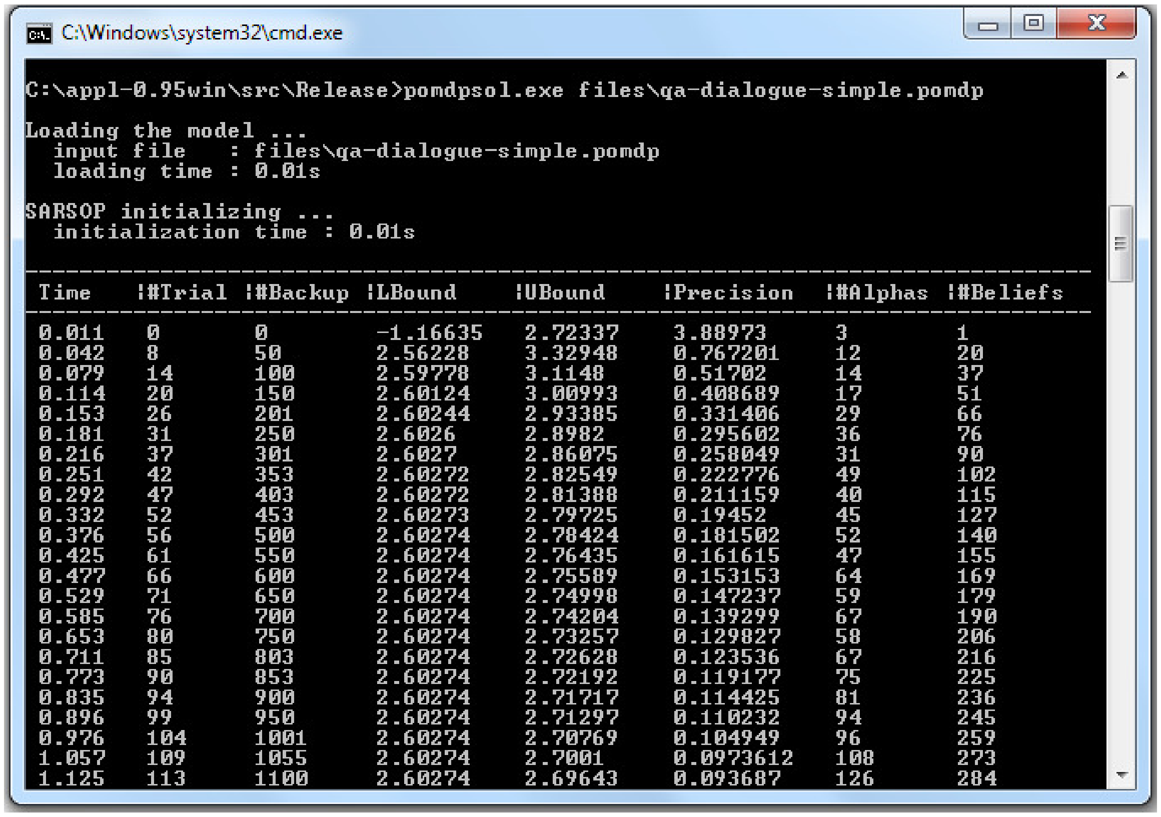
File  Edit  Format  View  Help

```
# version  1.7 PIZZA ORDER STATES (21 states).


discount: 0.95
values: reward
states: begin pizza  wings drinks p-reg p-veg w-hot w-buffalo d-coke d-pepsi p-cornmeal p-thin p-small p-medium p
actions: act-selectOnList act-ShowList act-YesNo
observations: pizza  type crust size  sauce toppings combo-drink wings w-sauce unknown happy sad angry
start: begin
#1      2       3       4       5       6       7       8       9       10      11      12      13      14      1
T:act-selectOnList:begin
0.01    0.33    0.33    0.33    0       0       0       0       0       0       0       0       0       0       0
T:act-selectOnList:pizza
0       0       0       0       0.25    0.25    0       0       0       0       0.25    0.25    0       0       0
T:act-selectOnList:wings
0       0       0       0       0       0       0.5     0.5     0       0       0       0       0       0       0
T:act-selectOnList:drinks
0       0       0       0       0       0       0       0       0.5     0.5     0       0       0       0       0
T:act-selectOnList:p-reg
0       0       0       0       0       0       0       0       0       0       0.5     0.5     0       0       0
T:act-selectOnList:p-veg
0       0       0       0       0       0       0       0       0       0       0.5     0.5     0       0       0
T:act-selectOnList:w-hot
0.33    0.33    0.01    0.33    0       0       0       0       0       0       0       0       0       0       0
T:act-selectOnList:w-buffalo
0.33    0.33    0.01    0.33    0       0       0       0       0       0       0       0       0       0       0
T:act-selectOnList:d-coke
0.5     0.2     0.2     0.1     0       0       0       0       0       0       0       0       0       0       0
T:act-selectOnList:d-pepsi
0.5     0.2     0.2     0.1     0       0       0       0       0       0       0       0       0       0       0
T:act-selectOnList:p-cornmeal
0       0       0       0       0       0       0       0       0       0       0       0       0.33    0.34    0
T:act-selectOnList:p-thin
0       0       0       0       0       0       0       0       0       0       0       0       0.33    0.34    0
T:act-selectOnList:p-small
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
T:act-selectOnList:p-medium
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
T:act-selectOnList:p-large
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
T:act-selectOnList:p-bbq
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
T:act-selectOnList:p-donair
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
T:act-selectOnList:p-tomatoe
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
T:act-selectOnList:p-chicken
0       0       0.5     0.5     0       0       0       0       0       0       0       0       0       0       0
T:act-selectOnList:p-pepperoni
0       0       0.5     0.5     0       0       0       0       0       0       0       0       0       0       0
T:act-selectOnList:p-greenpepper
0.1     0       0       0.9     0       0       0       0       0       0       0       0       0       0       0



#states: begin pizza  wings drinks p-reg p-veg w-hot w-buffalo d-coke d-pepsi p-cornmeal p-thin p-small p-medium
T:act-ShowList:begin
0       0.6     0.2     0.2     0       0       0       0       0       0       0       0       0       0       0
T:act-ShowList:pizza
0       0       0       0       0.5     0.5     0       0       0       0       0       0       0       0       0
T:act-ShowList:wings
0       0       0       0       0       0       0.5     0.5     0       0       0       0       0       0       0
T:act-ShowList:drinks
0       0       0       0       0       0       0       0       0.5     0.5     0       0       0       0       0
T:act-ShowList:p-reg
```

Figure 5.2.1.3: Tony Cassandra File Objects with 21 States

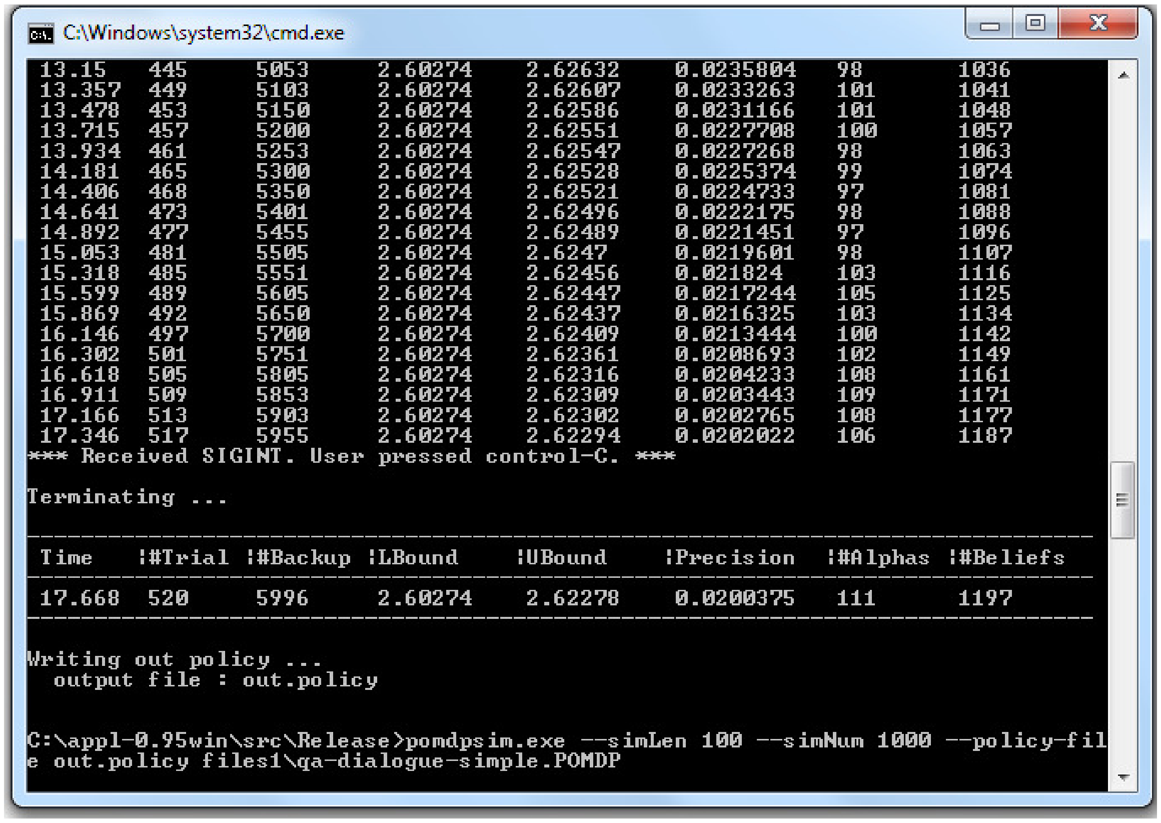
### 5.2.2 Generating Policy File

By receiving the POMDP specification file in Tony Cassandra's format, the ZMDP solver produces the *out.policy* file which specifies each action and state for selected POMDP file along with corresponding approximate optimal solution. In POMDP policy file, a set of lower bound values is set with an alpha vector and the corresponding actions are presented. With a current belief b, the lower bound on the expected long-term reward starting from b and that action leading to the expected lower bound can be known. In this experiment, the ZMDP solver was made to run for 17.66 minutes for the file specified and then stopped for generating the POMDP policy file.



```
C:\appl-0.95win\src\Release>pomdpsol.exe files\qa-dialogue-simple.pomdp

Loading the model ...
  input file   : files\qa-dialogue-simple.pomdp
  loading time : 0.01s

SARSOP initializing ...
  initialization time : 0.01s

-------------------------------------------------------------------------------
 Time   |#Trial |#Backup |LBound     |UBound     |Precision   |#Alphas |#Beliefs
-------------------------------------------------------------------------------
 0.011   0       0        -1.16635    2.72337     3.88973      3        1
 0.042   8       50       2.56228     3.32948     0.767201     12       20
 0.079   14      100      2.59778     3.1148      0.51702      14       37
 0.114   20      150      2.60124     3.00993     0.408689     17       51
 0.153   26      201      2.60244     2.93385     0.331406     29       66
 0.181   31      250      2.6026      2.8982      0.295602     36       76
 0.216   37      301      2.6027      2.86075     0.258049     31       90
 0.251   42      353      2.60272     2.82549     0.222776     49       102
 0.292   47      403      2.60272     2.81388     0.211159     40       115
 0.332   52      453      2.60273     2.79725     0.19452      45       127
 0.376   56      500      2.60274     2.78424     0.181502     52       140
 0.425   61      550      2.60274     2.76435     0.161615     47       155
 0.477   66      600      2.60274     2.75589     0.153153     64       169
 0.529   71      650      2.60274     2.74998     0.147237     59       179
 0.585   76      700      2.60274     2.74204     0.139299     67       190
 0.653   80      750      2.60274     2.73257     0.129827     58       206
 0.711   85      803      2.60274     2.72628     0.123536     67       216
 0.773   90      853      2.60274     2.72192     0.119177     75       225
 0.835   94      900      2.60274     2.71717     0.114425     81       236
 0.896   99      950      2.60274     2.71297     0.110232     94       245
 0.976   104     1001     2.60274     2.70769     0.104949     96       259
 1.057   109     1055     2.60274     2.7001      0.0973612    108      273
 1.125   113     1100     2.60274     2.69643     0.093687     126      284
```

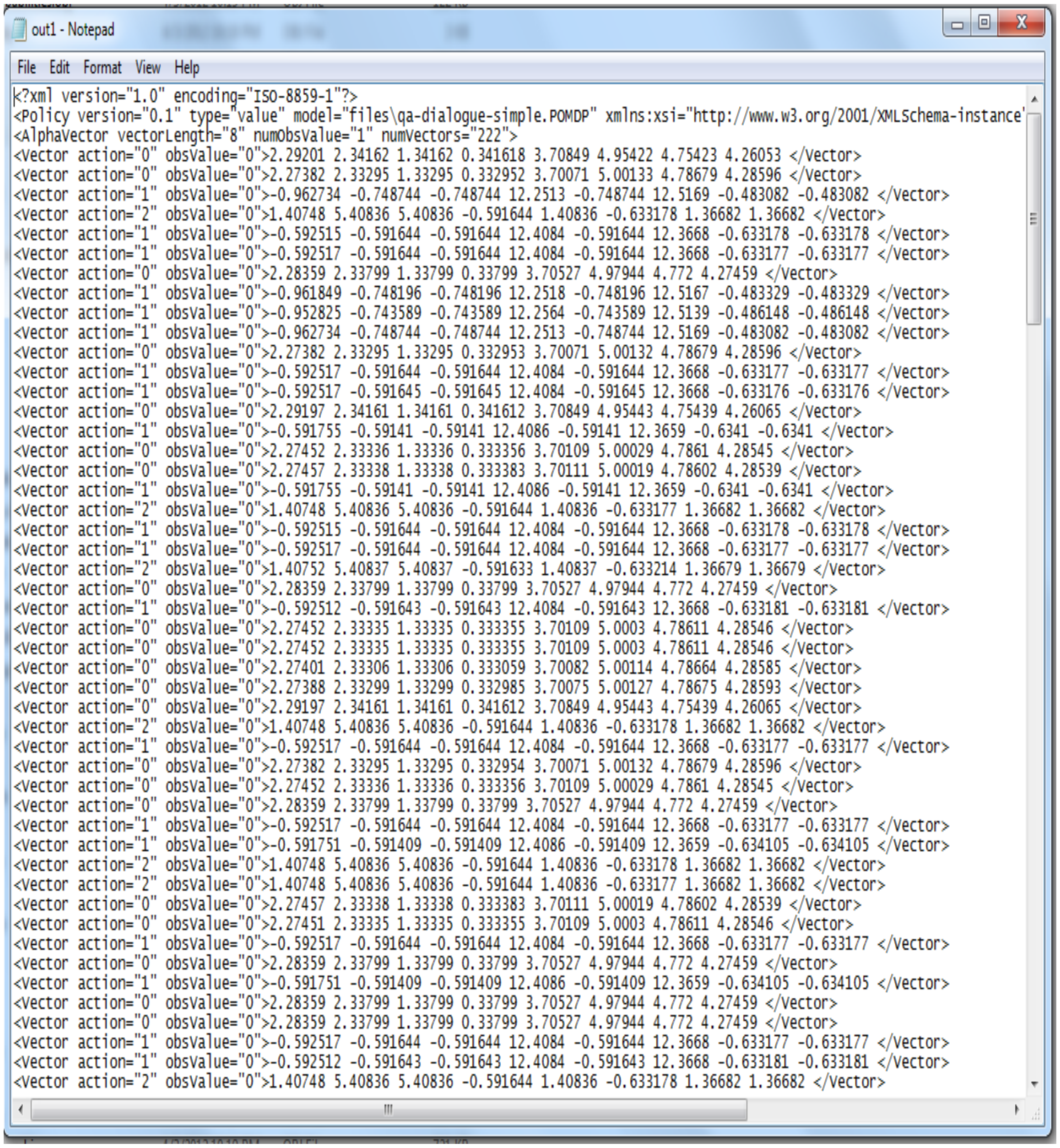
Figure 5.2.2.1: Generating Policy from POMDPSOL Binary File



```
C:\Windows\system32\cmd.exe
 13.15    445      5053      2.60274     2.62632     0.0235804    98        1036
 13.357   449      5103      2.60274     2.62607     0.0233263    101       1041
 13.478   453      5150      2.60274     2.62586     0.0231166    101       1048
 13.715   457      5200      2.60274     2.62551     0.0227708    100       1057
 13.934   461      5253      2.60274     2.62547     0.0227268    98        1063
 14.181   465      5300      2.60274     2.62528     0.0225374    99        1074
 14.406   468      5350      2.60274     2.62521     0.0224733    97        1081
 14.641   473      5401      2.60274     2.62496     0.0222175    98        1088
 14.892   477      5455      2.60274     2.62489     0.0221451    97        1096
 15.053   481      5505      2.60274     2.6247      0.0219601    98        1107
 15.318   485      5551      2.60274     2.62456     0.021824     103       1116
 15.599   489      5605      2.60274     2.62447     0.0217244    105       1125
 15.869   492      5650      2.60274     2.62437     0.0216325    103       1134
 16.146   497      5700      2.60274     2.62409     0.0213444    100       1142
 16.302   501      5751      2.60274     2.62361     0.0208693    102       1149
 16.618   505      5805      2.60274     2.62316     0.0204233    108       1161
 16.911   509      5853      2.60274     2.62309     0.0203443    109       1171
 17.166   513      5903      2.60274     2.62302     0.0202765    108       1177
 17.346   517      5955      2.60274     2.62294     0.0202022    106       1187
*** Received SIGINT. User pressed control-C. ***

Terminating ...

-------------------------------------------------------------------------------
 Time    |#Trial |#Backup |LBound     |UBound     |Precision  |#Alphas |#Beliefs
-------------------------------------------------------------------------------
 17.668  520      5996      2.60274     2.62278     0.0200375    111       1197
-------------------------------------------------------------------------------

Writing out policy ...
  output file : out.policy

C:\appl-0.95win\src\Release>pomdpsim.exe --simLen 100 --simNum 1000 --policy-fil
e out.policy files1\qa-dialogue-simple.POMDP
```

Figure 5.2.2.2: Terminating to Create the Output Policy File

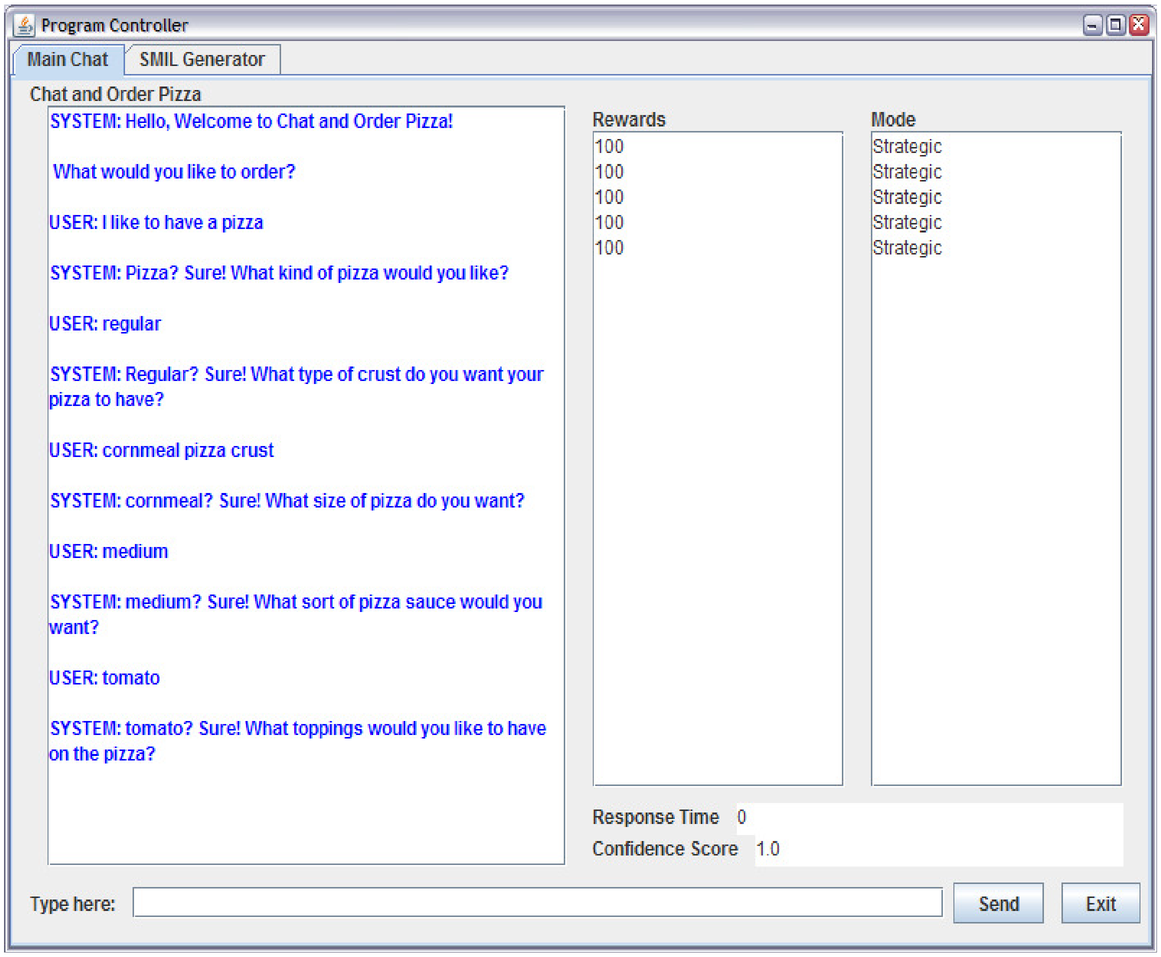
The generated policy file and the original POMDP file are given as inputs to the POMDP solver and the evaluator to make multiple runs and check the number of runs it takes to print the result. As well, we can utilize them to validate the POMDP state evaluation which is shown in Figure 5.2.2.3 below.



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Policy version="0.1" type="value" model="files\qa-dialogue-simple.POMDP" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance'
<AlphaVector vectorLength="8" numObsValue="1" numVectors="222">
<Vector action="0" obsValue="0">2.29201 2.34162 1.34162 0.341618 3.70849 4.95422 4.75423 4.26053 </Vector>
<Vector action="0" obsValue="0">2.27382 2.33295 1.33295 0.332952 3.70071 5.00133 4.78679 4.28596 </Vector>
<Vector action="1" obsValue="0">-0.962734 -0.748744 -0.748744 12.2513 -0.748744 12.5169 -0.483082 -0.483082 </Vector>
<Vector action="2" obsValue="0">1.40748 5.40836 5.40836 -0.591644 1.40836 -0.633178 1.36682 1.36682 </Vector>
<Vector action="1" obsValue="0">-0.592515 -0.591644 -0.591644 12.4084 -0.591644 12.3668 -0.633178 -0.633178 </Vector>
<Vector action="1" obsValue="0">-0.592517 -0.591644 -0.591644 12.4084 -0.591644 12.3668 -0.633177 -0.633177 </Vector>
<Vector action="0" obsValue="0">2.28359 2.33799 1.33799 0.33799 3.70527 4.97944 4.772 4.27459 </Vector>
<Vector action="1" obsValue="0">-0.961849 -0.748196 -0.748196 12.2518 -0.748196 12.5167 -0.483329 -0.483329 </Vector>
<Vector action="1" obsValue="0">-0.952825 -0.743589 -0.743589 12.2564 -0.743589 12.5139 -0.486148 -0.486148 </Vector>
<Vector action="1" obsValue="0">-0.962734 -0.748744 -0.748744 12.2513 -0.748744 12.5169 -0.483082 -0.483082 </Vector>
<Vector action="0" obsValue="0">2.27382 2.33295 1.33295 0.332953 3.70071 5.00132 4.78679 4.28596 </Vector>
<Vector action="1" obsValue="0">-0.592517 -0.591644 -0.591644 12.4084 -0.591644 12.3668 -0.633177 -0.633177 </Vector>
<Vector action="1" obsValue="0">-0.592517 -0.591645 -0.591645 12.4084 -0.591645 12.3668 -0.633176 -0.633176 </Vector>
<Vector action="0" obsValue="0">2.29197 2.34161 1.34161 0.341612 3.70849 4.95443 4.75439 4.26065 </Vector>
<Vector action="1" obsValue="0">-0.591755 -0.59141 -0.59141 12.4086 -0.59141 12.3659 -0.6341 -0.6341 </Vector>
<Vector action="0" obsValue="0">2.27452 2.33336 1.33336 0.333356 3.70109 5.00029 4.7861 4.28545 </Vector>
<Vector action="0" obsValue="0">2.27457 2.33338 1.33338 0.333383 3.70111 5.00019 4.78602 4.28539 </Vector>
<Vector action="1" obsValue="0">-0.591755 -0.59141 -0.59141 12.4086 -0.59141 12.3659 -0.6341 -0.6341 </Vector>
<Vector action="2" obsValue="0">1.40748 5.40836 5.40836 -0.591644 1.40836 -0.633177 1.36682 1.36682 </Vector>
<Vector action="1" obsValue="0">-0.592515 -0.591644 -0.591644 12.4084 -0.591644 12.3668 -0.633178 -0.633178 </Vector>
<Vector action="1" obsValue="0">-0.592517 -0.591644 -0.591644 12.4084 -0.591644 12.3668 -0.633177 -0.633177 </Vector>
<Vector action="2" obsValue="0">1.40752 5.40837 5.40837 -0.591633 1.40837 -0.633214 1.36679 1.36679 </Vector>
<Vector action="0" obsValue="0">2.28359 2.33799 1.33799 0.33799 3.70527 4.97944 4.772 4.27459 </Vector>
<Vector action="1" obsValue="0">-0.592512 -0.591643 -0.591643 12.4084 -0.591643 12.3668 -0.633181 -0.633181 </Vector>
<Vector action="0" obsValue="0">2.27452 2.33335 1.33335 0.333355 3.70109 5.0003 4.78611 4.28546 </Vector>
<Vector action="0" obsValue="0">2.27452 2.33335 1.33335 0.333355 3.70109 5.0003 4.78611 4.28546 </Vector>
<Vector action="0" obsValue="0">2.27401 2.33306 1.33306 0.333059 3.70082 5.00114 4.78664 4.28585 </Vector>
<Vector action="0" obsValue="0">2.27388 2.33299 1.33299 0.332985 3.70075 5.00127 4.78675 4.28593 </Vector>
<Vector action="0" obsValue="0">2.29197 2.34161 1.34161 0.341612 3.70849 4.95443 4.75439 4.26065 </Vector>
<Vector action="2" obsValue="0">1.40748 5.40836 5.40836 -0.591644 1.40836 -0.633178 1.36682 1.36682 </Vector>
<Vector action="1" obsValue="0">-0.592517 -0.591644 -0.591644 12.4084 -0.591644 12.3668 -0.633177 -0.633177 </Vector>
<Vector action="0" obsValue="0">2.27382 2.33295 1.33295 0.332954 3.70071 5.00132 4.78679 4.28596 </Vector>
<Vector action="0" obsValue="0">2.27452 2.33336 1.33336 0.333356 3.70109 5.00029 4.7861 4.28545 </Vector>
<Vector action="0" obsValue="0">2.28359 2.33799 1.33799 0.33799 3.70527 4.97944 4.772 4.27459 </Vector>
<Vector action="1" obsValue="0">-0.592517 -0.591644 -0.591644 12.4084 -0.591644 12.3668 -0.633177 -0.633177 </Vector>
<Vector action="1" obsValue="0">-0.591751 -0.591409 -0.591409 12.4086 -0.591409 12.3659 -0.634105 -0.634105 </Vector>
<Vector action="2" obsValue="0">1.40748 5.40836 5.40836 -0.591644 1.40836 -0.633178 1.36682 1.36682 </Vector>
<Vector action="2" obsValue="0">1.40748 5.40836 5.40836 -0.591644 1.40836 -0.633177 1.36682 1.36682 </Vector>
<Vector action="0" obsValue="0">2.27457 2.33338 1.33338 0.333383 3.70111 5.00019 4.78602 4.28539 </Vector>
<Vector action="0" obsValue="0">2.27451 2.33335 1.33335 0.333355 3.70109 5.0003 4.78611 4.28546 </Vector>
<Vector action="1" obsValue="0">-0.592517 -0.591644 -0.591644 12.4084 -0.591644 12.3668 -0.633177 -0.633177 </Vector>
<Vector action="0" obsValue="0">2.28359 2.33799 1.33799 0.33799 3.70527 4.97944 4.772 4.27459 </Vector>
<Vector action="1" obsValue="0">-0.591751 -0.591409 -0.591409 12.4086 -0.591409 12.3659 -0.634105 -0.634105 </Vector>
<Vector action="0" obsValue="0">2.28359 2.33799 1.33799 0.33799 3.70527 4.97944 4.772 4.27459 </Vector>
<Vector action="0" obsValue="0">2.28359 2.33799 1.33799 0.33799 3.70527 4.97944 4.772 4.27459 </Vector>
<Vector action="1" obsValue="0">-0.592517 -0.591644 -0.591644 12.4084 -0.591644 12.3668 -0.633177 -0.633177 </Vector>
<Vector action="1" obsValue="0">-0.592512 -0.591643 -0.591643 12.4084 -0.591643 12.3668 -0.633181 -0.633181 </Vector>
<Vector action="2" obsValue="0">1.40748 5.40836 5.40836 -0.591644 1.40836 -0.633178 1.36682 1.36682 </Vector>
```

Figure 5.2.2.3: Policy Output File

### 5.2.3 Test Case 1



Chat and Order Pizza

SYSTEM: Hello, Welcome to Chat and Order Pizza!

What would you like to order?

USER: I like to have a pizza

SYSTEM: Pizza? Sure! What kind of pizza would you like?

USER: regular

SYSTEM: Regular? Sure! What type of crust do you want your pizza to have?

USER: cornmeal pizza crust

SYSTEM: cornmeal? Sure! What size of pizza do you want?

USER: medium

SYSTEM: medium? Sure! What sort of pizza sauce would you want?

USER: tomato

SYSTEM: tomato? Sure! What toppings would you like to have on the pizza?

| Rewards | Mode |
|---|---|
| 100 | Strategic |
| 100 | Strategic |
| 100 | Strategic |
| 100 | Strategic |
| 100 | Strategic |

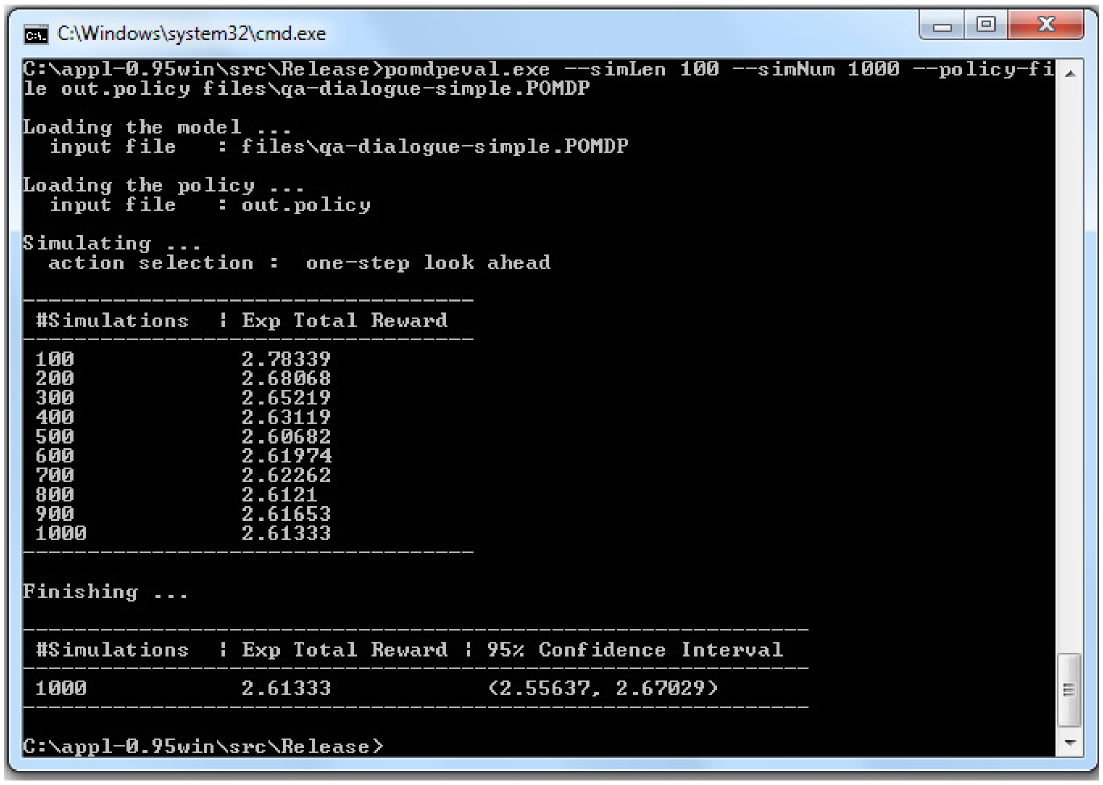
Response Time 0
Confidence Score 1.0

Figure 5.2.3.1: Test Case 1 in POMDP Mock up

We would set the above selected states and observations in the POMDP file and run the POMDP solver to predict the scenario with the user goal. The scenario outcome is as follows:

<Regular, medium pizza with cornmeal crust and tomato sauce>

This above situation is set as the user goal and we try to predict this using the solver and the results are as follows. The simulation only predicts when every user input is satisfactory.


C:\Windows\system32\cmd.exe

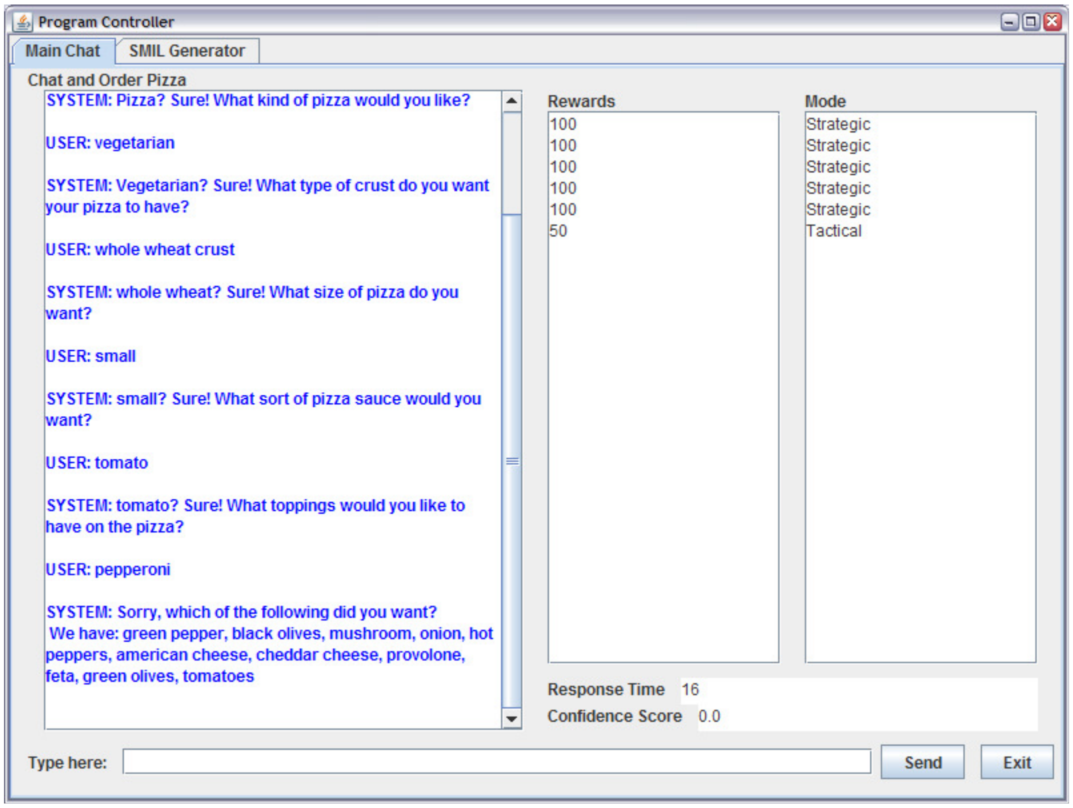
```
C:\appl-0.95win\src\Release>pomdpeval.exe --simLen 100 --simNum 1000 --policy-fi
le out.policy files\qa-dialogue-simple.POMDP

Loading the model ...
  input file   : files\qa-dialogue-simple.POMDP

Loading the policy ...
  input file   : out.policy

Simulating ...
  action selection :  one-step look ahead

-----------------------------------
 #Simulations  | Exp Total Reward
-----------------------------------
 100            2.78339
 200            2.68068
 300            2.65219
 400            2.63119
 500            2.60682
 600            2.61974
 700            2.62262
 800            2.6121
 900            2.61653
 1000           2.61333
-----------------------------------

Finishing ...

-------------------------------------------------------------
 #Simulations  | Exp Total Reward | 95% Confidence Interval
-------------------------------------------------------------
 1000           2.61333            (2.55637, 2.67029)
-------------------------------------------------------------

C:\appl-0.95win\src\Release>
```

Figure 5.2.3.2: Test Case 1 POMDP Simulation

| **Simulation** | **Output** |
|---|---|
| Total runs | 1000 |
| Number of times Scenario reached | 833 |
| Total turns | 4220 |
| Average turns per run | 4 |
| Standard deviation: | 3.882622246293481 |
| Average reward per turn | 1.7209302325581395 |

Table 5.2.3.3: Test Case 1 Results

### 5.2.4 Test Case 2



SYSTEM: Pizza? Sure! What kind of pizza would you like?

USER: vegetarian

SYSTEM: Vegetarian? Sure! What type of crust do you want your pizza to have?

USER: whole wheat crust

SYSTEM: whole wheat? Sure! What size of pizza do you want?

USER: small

SYSTEM: small? Sure! What sort of pizza sauce would you want?

USER: tomato

SYSTEM: tomato? Sure! What toppings would you like to have on the pizza?

USER: pepperoni

SYSTEM: Sorry, which of the following did you want?
We have: green pepper, black olives, mushroom, onion, hot peppers, american cheese, cheddar cheese, provolone, feta, green olives, tomatoes

| Rewards | Mode |
|---|---|
| 100 | Strategic |
| 100 | Strategic |
| 100 | Strategic |
| 100 | Strategic |
| 100 | Strategic |
| 50 | Tactical |

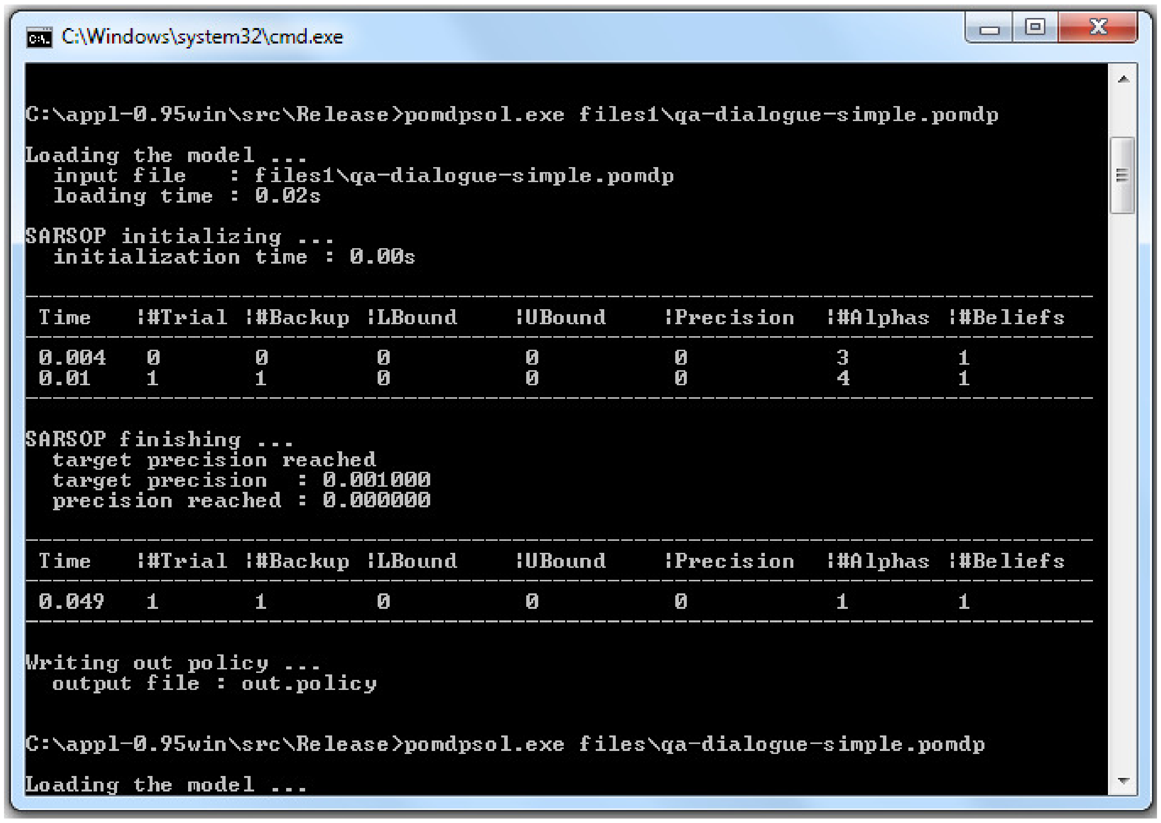
Response Time 16
Confidence Score 0.0

Figure 5.2.4.1: Test Case 2 in POMDP Mock up

In this case we would set the above selected states and observations in the POMDP file and run the POMDP solver to predict the scenario with the user goal. The scenario here is as follows:

<Regular, small pizza with whole wheat crust and tomato sauce and topped with pepperoni>

This above situation is set as the user goal. We try to predict this using the solver, but the system does not arrive at a solution as the simulation only predicts when every user input is appropriate. It is shown in Figure 5.2.4.2.

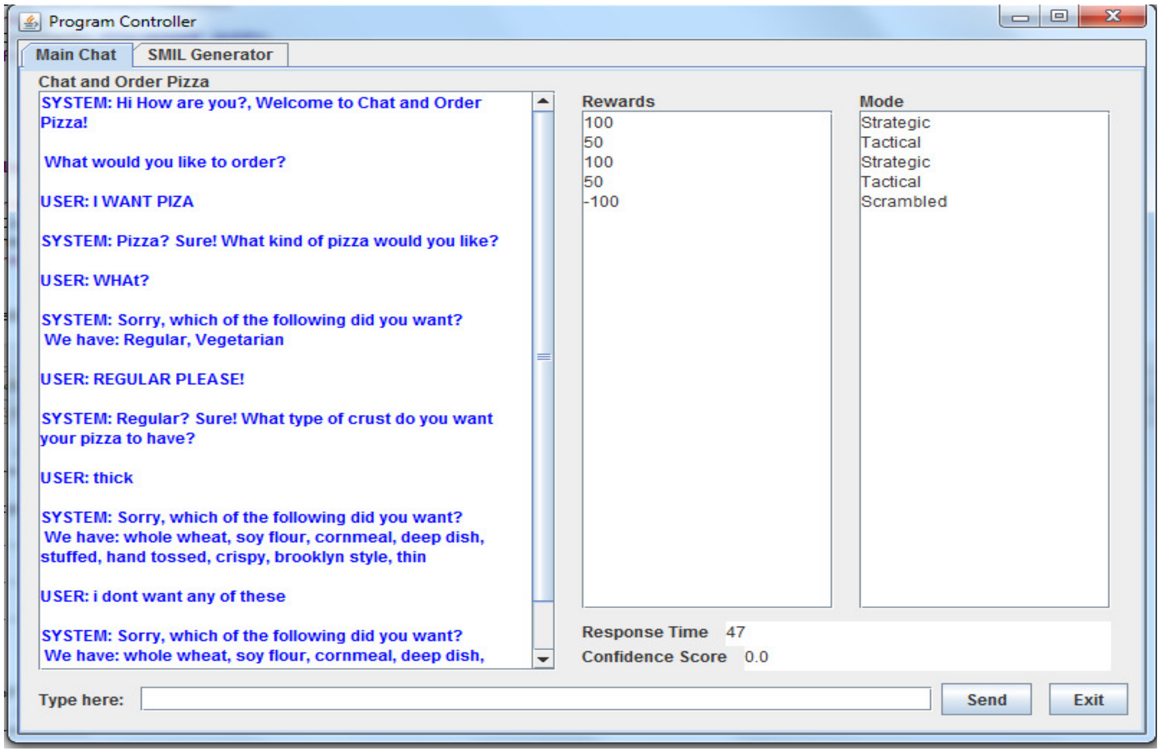


```
C:\appl-0.95win\src\Release>pomdpsol.exe files1\qa-dialogue-simple.pomdp

Loading the model ...
  input file   : files1\qa-dialogue-simple.pomdp
  loading time : 0.02s

SARSOP initializing ...
  initialization time : 0.00s

-------------------------------------------------------------------------------
 Time   |#Trial |#Backup |LBound    |UBound    |Precision  |#Alphas |#Beliefs
-------------------------------------------------------------------------------
 0.004   0       0        0          0          0           3        1
 0.01    1       1        0          0          0           4        1
-------------------------------------------------------------------------------

SARSOP finishing ...
  target precision reached
  target precision  : 0.001000
  precision reached : 0.000000

-------------------------------------------------------------------------------
 Time   |#Trial |#Backup |LBound    |UBound    |Precision  |#Alphas |#Beliefs
-------------------------------------------------------------------------------
 0.049   1       1        0          0          0           1        1
-------------------------------------------------------------------------------

Writing out policy ...
  output file : out.policy

C:\appl-0.95win\src\Release>pomdpsol.exe files\qa-dialogue-simple.pomdp

Loading the model ...
```

Figure 5.2.4.2: Test Case 2 POMDP Simulation

Since there exists a conflict in the user input, the POMDP does not solve the policy as the belief updated or the initial belief has a value to a meat toppings for a pizza whereas the state chosen earlier is ‘p-veg’ which is a vegetarian pizza. Hence there is only the first line written in the policy file, which gives out an error when we try to simulate.

### 5.2.5 Test Case 3



```
SYSTEM: Hi How are you?, Welcome to Chat and Order Pizza!

 What would you like to order?

USER: I WANT PIZA

SYSTEM: Pizza? Sure! What kind of pizza would you like?

USER: WHAt?

SYSTEM: Sorry, which of the following did you want?
 We have: Regular, Vegetarian

USER: REGULAR PLEASE!

SYSTEM: Regular? Sure! What type of crust do you want your pizza to have?

USER: thick

SYSTEM: Sorry, which of the following did you want?
 We have: whole wheat, soy flour, cornmeal, deep dish, stuffed, hand tossed, crispy, brooklyn style, thin

USER: i dont want any of these

SYSTEM: Sorry, which of the following did you want?
 We have: whole wheat, soy flour, cornmeal, deep dish,
```

| Rewards | Mode |
|---|---|
| 100 | Strategic |
| 50 | Tactical |
| 100 | Strategic |
| 50 | Tactical |
| -100 | Scrambled |

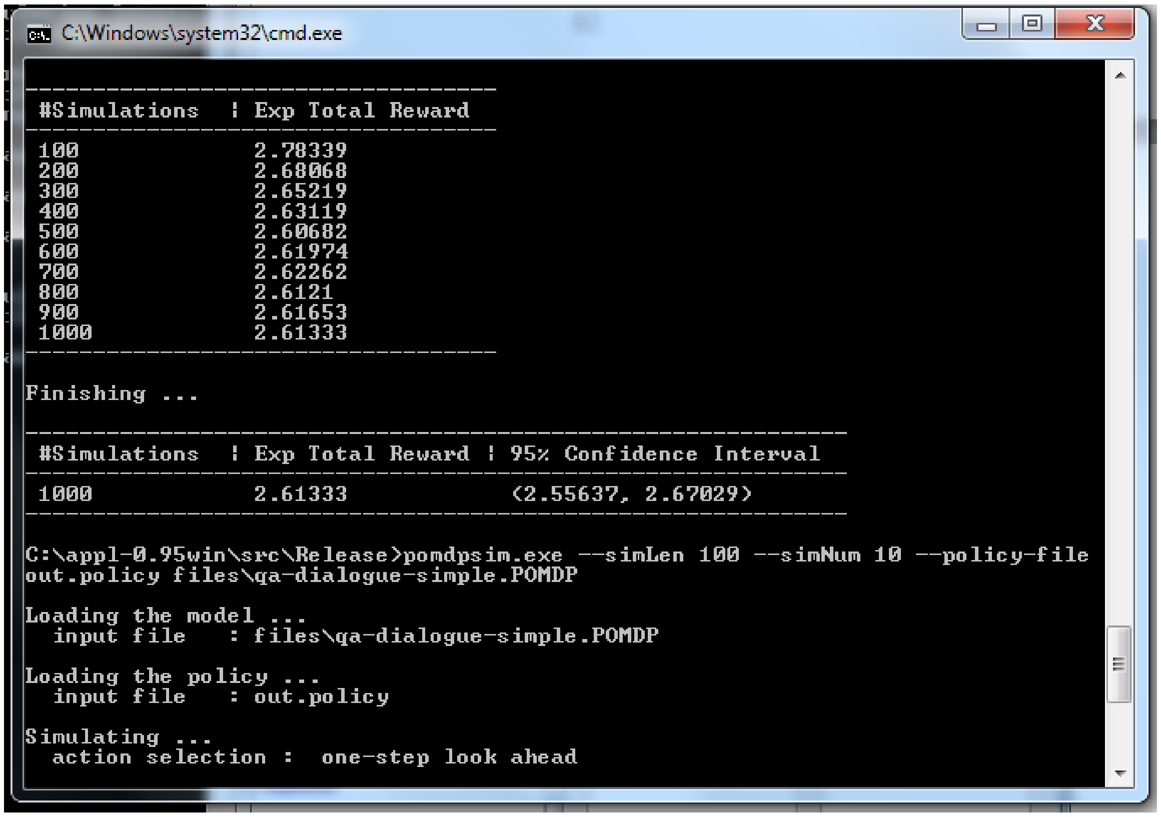
Response Time 47
Confidence Score 0.0

Figure 5.2.5.1: Test Case 3 in POMDP Mock up

In the above specified case, the user has become angry in emotion since he did not get a ‘thick’ crust as it is not there in the menu. The system will have to run through the information space to get the possible result which is the maximum number of previously returned scenario and try acting randomly to see if the user wants the same.

| Simulation | Output |
|---|---|
| Total runs | 10 |
| Number of times Scenario reached | 9 |
| Total turns | 2 |
| Average turns per run | 2 |
| Standard deviation: | 0.96978843211 |
| Average reward per turn | 2.6 |

Table 5.2.5.2: Test Case 3 Results



```
-----------------------------------
 #Simulations   | Exp Total Reward
-----------------------------------
 100             2.78339
 200             2.68068
 300             2.65219
 400             2.63119
 500             2.60682
 600             2.61974
 700             2.62262
 800             2.6121
 900             2.61653
 1000            2.61333
-----------------------------------

Finishing ...

----------------------------------------------------------------
 #Simulations   | Exp Total Reward | 95% Confidence Interval
----------------------------------------------------------------
 1000            2.61333            (2.55637, 2.67029)
----------------------------------------------------------------

C:\appl-0.95win\src\Release>pomdpsim.exe --simLen 100 --simNum 10 --policy-file
out.policy files\qa-dialogue-simple.POMDP

Loading the model ...
  input file   : files\qa-dialogue-simple.POMDP

Loading the policy ...
  input file   : out.policy

Simulating ...
  action selection :  one-step look ahead
```

Figure 5.2.5.3: The 10 runs for the Output File

The simulated results from the mock up are as follows:

```
Emotion: Pleased

USER: I WANT PIZA

------------------------------

Rewards: 100

------------------------------

Confidence: 1.0

++++++++++++++++++++++++++++++


Emotion: Displeased

USER: WHAT?

------------------------------

Rewards: 50

------------------------------

Confidence: 0.0

++++++++++++++++++++++++++++++

Emotion: Pleased

USER: REGULAR PLEASE!

------------------------------

Rewards: 100

------------------------------

Confidence: 1.0
```

```
++++++++++++++++++++++++++++++

Emotion: Displeased

USER: thick

------------------------------

Rewards: 50

------------------------------

Confidence: 0.0


++++++++++++++++++++++++++++++

Emotion: Fear

USER: i dont want any of these

------------------------------

Rewards: -100

------------------------------

Confidence: 0.0
```

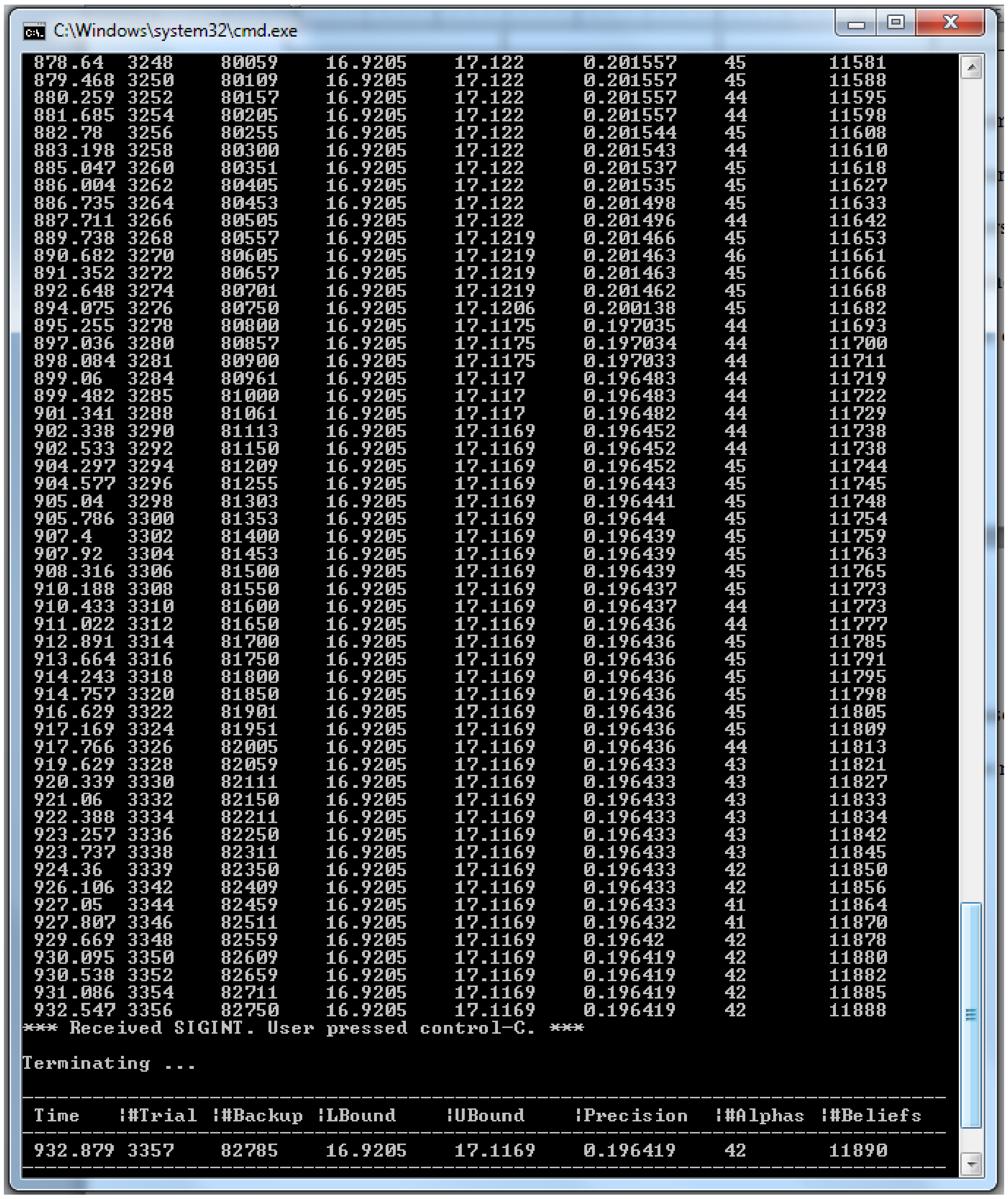
## 5.3 Results and Discussion

We simulated five different test cases to analyse the performance of our new approach. The above specified test cases demonstrate the efficiency of the new improved Modified POMDP in comparison to the previous work, which are shown in Table 5.3.2, where we have compared the results with the benchmarks and baseline works for 1000 runs for a total of about 233 similar scenarios. If the system achieves higher confidence scores it means that it can understand the user utterances better in a regular or noisy environment.

| | *Baseline 1[7]* *Intention level only* | *Baseline 2[6]* *Intention level only* | *Proposed Work* *Intention + Emotion* |
|---|---|---|---|
| Average Number of Turns for scenarios | 8 | 11 | 5 |
| Average number of times scenario achieved per 1000 turns | 712 | 745 | 822 |
| Accuracy (percentage) | 71 | 75 | 82 |
| Standard Deviation | 5.4 | 4.6 | 3.8 |

Table 5.3.1: Comparison Results

The above table infers that it outperforms the two baseline works in performance, efficiency and standard deviation. The baselines' dialogues have higher standard deviation given that the proportion of number of turns per dialogue is more disperse. The dialogue gathered in the new modified approach has a smaller deviation since the successful dialogues are usually those which require the minimum number of turns to achieve the objective which is the user goal. This is also basically because of the addition of emotion values in the observation as it makes the history space more structured.



```
C:\Windows\system32\cmd.exe
 878.64  3248     80059    16.9205     17.122     0.201557    45        11581
 879.468 3250     80109    16.9205     17.122     0.201557    45        11588
 880.259 3252     80157    16.9205     17.122     0.201557    44        11595
 881.685 3254     80205    16.9205     17.122     0.201557    44        11598
 882.78  3256     80255    16.9205     17.122     0.201544    45        11608
 883.198 3258     80300    16.9205     17.122     0.201543    44        11610
 885.047 3260     80351    16.9205     17.122     0.201537    45        11618
 886.004 3262     80405    16.9205     17.122     0.201535    45        11627
 886.735 3264     80453    16.9205     17.122     0.201498    45        11633
 887.711 3266     80505    16.9205     17.122     0.201496    44        11642
 889.738 3268     80557    16.9205     17.1219    0.201466    45        11653
 890.682 3270     80605    16.9205     17.1219    0.201463    46        11661
 891.352 3272     80657    16.9205     17.1219    0.201463    45        11666
 892.648 3274     80701    16.9205     17.1219    0.201462    45        11668
 894.075 3276     80750    16.9205     17.1206    0.200138    45        11682
 895.255 3278     80800    16.9205     17.1175    0.197035    44        11693
 897.036 3280     80857    16.9205     17.1175    0.197034    44        11700
 898.084 3281     80900    16.9205     17.1175    0.197033    44        11711
 899.06  3284     80961    16.9205     17.117     0.196483    44        11719
 899.482 3285     81000    16.9205     17.117     0.196483    44        11722
 901.341 3288     81061    16.9205     17.117     0.196482    44        11729
 902.338 3290     81113    16.9205     17.1169    0.196452    44        11738
 902.533 3292     81150    16.9205     17.1169    0.196452    44        11738
 904.297 3294     81209    16.9205     17.1169    0.196452    45        11744
 904.577 3296     81255    16.9205     17.1169    0.196443    45        11745
 905.04  3298     81303    16.9205     17.1169    0.196441    45        11748
 905.786 3300     81353    16.9205     17.1169    0.19644     45        11754
 907.4   3302     81400    16.9205     17.1169    0.196439    45        11759
 907.92  3304     81453    16.9205     17.1169    0.196439    45        11763
 908.316 3306     81500    16.9205     17.1169    0.196439    45        11765
 910.188 3308     81550    16.9205     17.1169    0.196437    45        11773
 910.433 3310     81600    16.9205     17.1169    0.196437    44        11773
 911.022 3312     81650    16.9205     17.1169    0.196436    44        11777
 912.891 3314     81700    16.9205     17.1169    0.196436    45        11785
 913.664 3316     81750    16.9205     17.1169    0.196436    45        11791
 914.243 3318     81800    16.9205     17.1169    0.196436    45        11795
 914.757 3320     81850    16.9205     17.1169    0.196436    45        11798
 916.629 3322     81901    16.9205     17.1169    0.196436    45        11805
 917.169 3324     81951    16.9205     17.1169    0.196436    45        11809
 917.766 3326     82005    16.9205     17.1169    0.196436    44        11813
 919.629 3328     82059    16.9205     17.1169    0.196433    43        11821
 920.339 3330     82111    16.9205     17.1169    0.196433    43        11827
 921.06  3332     82150    16.9205     17.1169    0.196433    43        11833
 922.388 3334     82211    16.9205     17.1169    0.196433    43        11834
 923.257 3336     82250    16.9205     17.1169    0.196433    43        11842
 923.737 3338     82311    16.9205     17.1169    0.196433    43        11845
 924.36  3339     82350    16.9205     17.1169    0.196433    42        11850
 926.106 3342     82409    16.9205     17.1169    0.196433    42        11856
 927.05  3344     82459    16.9205     17.1169    0.196433    41        11864
 927.807 3346     82511    16.9205     17.1169    0.196432    41        11870
 929.669 3348     82559    16.9205     17.1169    0.19642     42        11878
 930.095 3350     82609    16.9205     17.1169    0.196419    42        11880
 930.538 3352     82659    16.9205     17.1169    0.196419    42        11882
 931.086 3354     82711    16.9205     17.1169    0.196419    42        11885
 932.547 3356     82750    16.9205     17.1169    0.196419    42        11888
*** Received SIGINT. User pressed control-C. ***

Terminating ...

-------------------------------------------------------------------------------
 Time    |#Trial |#Backup |LBound     |UBound     |Precision  |#Alphas |#Beliefs
-------------------------------------------------------------------------------
 932.879 3357     82785    16.9205     17.1169    0.196419    42        11890
-------------------------------------------------------------------------------
```

Figure 5.3.2: POMDP Values with Multiple Scenarios

### 5.3.1 Analysis of Emotion Recognition

We achieved better accuracy with the validation experiment. Classifying sentences with the emotion as one of the six primary emotions are included in Table 5.3.1.1 and compared with the two tuning conditions on the main feature sets and a baseline. This clearly reveals the accuracy of the customized decision tree algorithm with comparison to the other two baselines mentioned.

| *Method* | *Average Accuracy* |
|---|---|
| Hierarchical Classification BoW+SO[20] | 55.24% |
| All features + sequencing(same-tune-eval)[16] | 69.37% |
| All features + sequencing(sep-tune-eval)[16] | 62.94% |
| Customized decision tree algorithm[1] | 84.36% |

Table 5.3.1.1: CDT Accuracy Comparison Table [1]

The intensity evaluation used in the system works well with the generic phrases and statements and helps by being homogenous throughout the learning process of the system. However it could be modified by including a clause to evaluate more specific statements that have direct forms of emotional keywords such as ‘hate’, ‘happy’ and so on.

## 5.4 Conclusion

This chapter examined the efficiency and effectiveness of the new proposed modified POMDP system with emotions. The emotion recognition process using Customized Decision Tree (CDT) algorithm has showed better results than some of the existing works.  The qualitative analysis supports that the proposed system is reliable with more capabilities compared to some of the baselines and other works specified. The quantitative analysis shows better results which were carried by simulating different test cases. This thesis has laid the baseline for adding emotions into the POMDP based model and given a perspective of why it is important in improving the user intention discovery technique. Some of the future work and recommendations directed towards this area of human-computer interaction can be found in the next upcoming chapter.

# CHAPTER VI

# CONCLUSION AND FUTURE WORKS

In this thesis, the main dialogue management approaches are observed under the pizza ordering based domain agent. As well, the history information space theory is discussed and a detailed investigation of the major approaches of dialogue management methodologies with the philosophy of information space reveals reasons for their problems. With the analysis, the problem of the existing POMDP based approach is identified which we conclude by saying it is more efficient in intention discovery while giving the user emotion as the input. The Markovian model over the belief state in the dialogue management process is challenged because it loses some noteworthy information needed for decision making. Therefore, the original POMDP-based approach applied in the dialogue management cannot detect uncertainties in the belief state which are caused by the domain knowledge constraints. Based on the theory, a modified approach is proposed to enable POMDP-based dialogue management to handle uncertainties in belief state itself by giving the user emotions and also directing the history information space. Experimental results demonstrate significant improvement by the new approach towards accurate recognition of the user's intention. The advantage is more obvious when it comes with the scenario that user has lack of knowledge and provides unreasonable information to the agent. Instead, the process still tries to suggest the user with essential possible scenario. For the future work, active investigation is under way to include the changing trend of belief state in the process of planning for the construction of a real straight, applicable, vibrant, and instructive dialogue structure. As well, another important direction is that to investigate the more practical model to solve the POMDP based approach scale up problem. When the domain is complicated, the state space of POMDP specification file can be certainly massive and the POMDP's elucidation is reckoning exorbitant. The current active researches have already put lots of efforts in this area to design more practical background and POMDP solution procedure to speed up the approximate solution finding process. We can further import the system to mobile application by applying mobile computing techniques in the system. More importantly this approach utilizes discrete probability set to provide the belief state and update it which could be transformed to more of a continuous form which is the holy grail of robot communication or multi-agent systems. Lastly, we assume that emotion handling is enough to produce user intentions but it has definitely opened up a whole set of questions such as improving the system by giving a mathematical human physiological model to the agent to produce efficient results. As well, since the POMDP approach by itself is a computationally complex method, we can try to hybridize it with another agent technique for example: BDI-POMDP approach which has a lot of potential over multi-agent gaming systems with user/agent emotions. These types of developments in the field of dialogue management will dominate the world's technology by using avatars and robots to act much more naturally in providing support to the human beings.

## REFERENCES

[1] Sivaraman Sriram, Xiaobu Yuan, "An Enhanced Approach for Classifying Emotions using Customized Decision Tree Algorithm", IEEE Southeastcon 2012, Data Mining and Machine Learning, March 15-18, 2012, pp. 1-6

[2] Myers, David G. “Theories of Emotion.” Psychology: Seventh Edition, Worth Publishers, New York, 2004.

[3] Yong-Soo seol, Dong-Joo Kim and Han-Woo Kim, “Emotion Recognition from Text using Knowledge-based ANN”, The 23rd International Technical Conference on Circuit/Systems, Computers and Communications. (ITS-CSCC-2008), pp. 1569-1572

[4] Jun Wong, J., Yeung Cho, S. “A local experts organization model with application to face emotion recognition”, Expert systems with application, 2008, pp. 804-819

[5]Alfred S. Gilman, “Universal Design and the Grid”, an article from the Trace Center, College of Engineering, University of Winsconsin-Madison, 2010

[6] Xiaobu Yuan, Libian Bian (2010), "A modified approach of POMDP-based dialogue management ", Robotics and Biomimetics - ROBIO, 2010, pp. 816-821

[7] Sathulla Sabiha, Four Mode Based Dialogue Management with Modified POMDP Model, MSc Thesis, School of Computer Science University of Windsor, 2011, pp. 24-29.

[8] Chunling Ma, Helmut Prendinger, and Mitsuru Ishizuka, “A Chat System Based on Emotion Estimation from Text and Embodied Conversational Messengers”, Graduate School of Information Science and Technology, University of Tokyo and National Institute of Informatics, Japan. 2006, pp. 535-538

[9] E. Leon, G. Clarke, F. Sepulveda, V. Callaghan, 2004, “Optimised attribute selection for emotion classification using physiological signals”. Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, San Francisco, CA, pp. 184-187.

[10] Georgeff, M. P., Pell, B., Pollack, M. E., Tambe, M. and Wooldridge, M. (1999): The belief-desire-intention model of agency. In: Proceedings of the 5th International Workshop on Intelligent Agents, eds. Müller, J. P., Singh, M. P. and Rao, A. S., LNAI Vol. 1555, Springer, pp. 1-10

[11] M. E. Bratman, “Intentions, Plans, and Practical Reason”, An Article from Harvard University Press: Cambridge, MA, 1987, pp. 2-7

[12] Schwitzgebel, Eric, “Belief”. An article taken from the resource provided by the Stanford Encyclopaedia of Philosophy, August 14th, 2006, http://plato.stanford.edu/entries/belief/

[13] C. Brown, “What is a Belief state”, *Midwest Studies in Philosophy,* http://www.trinity.edu/cbrown/papers/, 2002, pp. 357- 378,

[14] Cheng-Yu Lu, Jen-Shin Hong and Samuel Cruz-Lara, “Emotion Detection in Textual Information by Semantic Role Labeling and Web Mining Techniques”, LORIA (UMR 7503) – University of Nancy. 2006, pp. 7-11

[15] C. Wu, Z.Chuang, and Y. Lin, “ Emotion Recognition from Text using Semantic Labels and Seperable Mixture Models,’’ ACM Transactions on Asian Language Information Processing, Vol.5, No. 2, June 2006, pp. 168-182

[16] Jun Wong, J., Yeung Cho, S. “A local experts organization model with application to face emotion recognition”, Expert systems with application, 2008, pp. 804-819

[17] Cheng, S. C. “Semantic- based facial expression recognition using analytical hierarchy process”, Expert systems with applications, 2007, pp. 86-95

[18] A. Jamshidined, “Facial Emotion Recognition for Human Computer Interaction Using a Fuzzy Model in the E-Business”, 2009 pp. 202–204,

[19] Castellano, G., Villalba, S., & Camurri, A, “Recognizing human emotions from body movement and gesture dynamics”, *Affective Computing and Intelligent Interaction, ACII 2007, September 2007,* pp. 71-82

[20] A. Egges, S. Kshirsagar, N. M.Thalmann, “Generic personality and emotion simulation for conversational agents”, Computer Animation and Virtual Worlds, Vol.15, 2004, pp.1-13.

[21] L. Devillers, I. Vasilescu, and L. Lamel, “Annotation and Detection of Emotion in a task-oriented human–human dialog corpus”, In Proceedings of ISLE Workshop on Dialogue Tagging for Multi-Modal Human Computer Interaction, pp. 15-17, 2002.

[22] M. Gasic and S. Young (2011). "Effective Handling of Dialogue State in the Hidden Information State POMDP-based Dialogue Manager." ACM Transactions on Speech and Language Processing, 7(3), pp. 1-25

[23] F. Jurcicek, B. Thomson and S. Young (2012), "Reinforcement learning for parameter estimation in statistical spoken dialogue systems", Computer Speech and Language, 26(3):pp. 127-228

[24] LB Larsen and A. Baekgaard. Rapid prototyping of a dialog system using a generic dialog development platform. In Proc ICSLP International Conference on Spoken Language Processing, Yokohama, 1994, pp. 919-922

[25] P. Cohen. Dialogue modelling. 1996): Survey of the State of the Art in Human Language Technology. Oregon Graduate Institute of Technology, http://www. cse. ogi.edu/CSLU/HLTsurvey, 1996.

[26] M.F. McTear. Spoken dialogue technology: enabling the conversational user interface. ACM Computing Surveys (CSUR), 34(1): 2002, pp. 169-172

[27] Wayne Ward, Bryan Pellom, Xiuyang Yu, Kadri Hacioglu, "Improvements in Audio Processing and Language Modeling in the CU Communicator", Eurospeech 2001, Aalborg Denmark, Sept. 2001, pp. 187-192

[28]N. Dahlback, A. Flycht-Eriksson, A. Jonsson, and P. Qvarfordt. An architechture for multi-modal natural dialogue systems. In ESCA Tutorial and Research Workshop (ETRW) on Interactive Dialogue in Multi-Modal Systems, 1999, pp. 53-56

[29] D. Goddeau, H. Meng, J. Polifroni, S. Seneff, and S. Busayapongchai. A form-based dialogue manager for spoken language applications. In Fourth International Conference on Spoken Language Processing, volume 2, pp. 701-704, 1996.

[30] D. Bohus and A.I. Rudnicky. RavenClaw: Dialog management using hierarchical task decomposition and an expectation agenda. In Eighth European Conference on Speech Communication and Technology. ISCA, 2003, pp. 332-361

[31]A.I. Rudnicky, E. Thayer, P. Constantinides, C. Tchou, R. Shern, K. Lenzo, W. Xu, and A. Oh, "Creating natural dialogs in the Carnegie Mellon Communicator system", In Sixth European Conference on Speech Communication and Technology, ESCA, 1999, pp. 1531-1534

[32] J.H. Wright, A.L. Gorin, and A. Abella, "Spoken language understanding within dialogs using a graphical model of task structure", In Fifth International Conference on Spoken Language Processing, volume 5, ECSLP, 1998, pp. 1879-1882

[33] M. Denecke and A. Waibel, "Dialogue strategies guiding users to their communicative goals", In Proc. Eurospeech, volume 96, 1997, pp. 265-274

[34] Jason D. Williams, "Exploiting the ASR N-Best by tracking multiple dialog state hypotheses", Proc Interspeech, Brisbane, Australia, 2008, pp. 191-194

[35] S. Young, "Using POMDPs for Dialog Management", IEEE/ACL Workshop on Spoken Language Technology, Aruba, 2006, pp.177-183

[36] MC Kim, PH Seong, E Hollnagel, "A probabilistic approach for determining the control mode in CREAM", Reliability Engineering & System Safety, Volume 91, Issue 2, 2006, pp. 191-199.

[37] N A Stanton, M J Ashleigh, A D Roberts, F Xu. Testing Hollnagel's Contextual Control Model: Assessing team behavior in a human Supervisory control task. International Journal of Cognitive Ergonomics, 2001, pp. 679-694

[38] Libian Bian, A Historical Information approach of POMDP-BASED Dialogue Management, MSc Thesis, School of Computer Science University of Windsor, 2010.

[39] Augusto Cesar Espíndola Baffa, Angelo E. M. Ciarlini: Modeling POMDPs for generating and simulating stock investment policies, SAC 2010: pp. 2394-2399

[40] Nils J Nilsson, "Introduction to Machine Learning, An early draft", Department of Computer Science, Stanford University, 4th December 1996, pp. 1-26

[41] Wei Peng, Juhua Chen and Haiping Zhou, " An Implementation of ID3- Decision Tree Learning Algorithm, n Article of Project Report, School of Computer Science & Engineering, University of New South Wales, Australia, 2010, pp. 8-23

[42] Thelwall, M., Buckley, K., & Paltoglou, G. (2012), "Sentiment strength detection for the social Web", *Journal of the American Society for Information Science and Technology*, 63(1), pp. 163-173.

[43] Carlo Strapparava, Rada Mihalcea SemEval-2007 Task 14: Affective Text FBK – first Istituto per la Ricerca Scientica e Tecnologica I-38050, Povo, Trento, Italy Department of Computer Science University of North Texas Denton, TX, 76203, USA

[44] Marc Batchelor and Jens Bleuel, "Latest Pentaho Data Integration Documentation",http://wiki.pentaho.com/display/EAI/Latest+Pentaho+Data+Integration+%28aka+Kettle%29+Documentation; November, 2011.

[45] Tony Cassandra. Tony's POMDP page, http://www.cassandra.org/pomdp/index.shtml, May 2010.

[46] Trung H. Bui, Dennis Hofs, and Boris van Schooten. Dialogue specification parser.http://wwwhoms.cs.utwente.nl/"hofs/porndp/, May 2010.

[47] T. Smith, "ZMDP software for POMDP and MDP planning", http://www.cs.emu.edu/trey/zmdp/. May 2010.

[48] H. Kurniawati and W. S. Lee. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces, Robotics: 2008, pp.12-23

# VITA AUCTORIS

Sivaraman Sriram was born in 1986 in Chennai, India. He graduated from Anna University where he obtained a B.Tech in Information Technology in 2008. He is currently a candidate for the Master's degree in Computer Science at the University of Windsor and hopes to graduate in Spring 2012.