Framework for Human Computer Interaction for Learning Dialogue Strategies using Controlled Natural Language in Information Systems

A thesis submitted in partial fulfilment of the
requirement for the degree of

Doctor of Philosophy (Ph.D)

in

Computer Science

by

Manzoor Ahmad

under the supervision of

Dr. S.M.K. Quadri

P.G. Department of Computer Sciences
Faculty of Applied Sciences & Technology
University of Kashmir

July, 2012

Declaration

This is to certify that the thesis entitled **"Framework for Human Computer Interaction for learning dialogue strategies using controlled natural language in information systems"**, submitted by **Mr. Manzoor Ahmad** in the *Department of Computer Sciences, University of Kashmir, Srinagar* for the award of the degree of **Doctor of Philosophy** in **Computer Science**, is a record of an original research work carried out by him under my supervision and guidance without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. The thesis has fulfilled all the requirements as per the regulations of the University and in my opinion has reached the standards required for the submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma

Supervisor and Head
(Dr. S.M.K. Quadri)
Department of Computer Sciences
University of Kashmir

Dated: 13-July-2012

To my father and mother,

for their continuous love, encouragement,care and support.

Acknowledgements

First of all, I would like to sincerely thank almighty ”ALLAH” for all his grants that he bestowed on me,given me a chance and courage to complete this Ph.D. Thesis.

I would like to thank to my PhD Supervisor, Dr.S.M.K.Quadri who has been an incredible help throughout my research period, encouraging, lively, enthusiastic, and energetic, would always keep me focussed with my research by his timely lectures on research methodology. Dr Quadri has been supportive and has given me the freedom to think more independently about our experiments and results.He has also provided insightful discussions about the research. I am also very grateful to Late Dr. Mehraj-u-Din Dar for his scientific advice, knowledge and many insightful discussions and suggestions. He was my primary source of inspiration and was instrumental in my completion of this thesis.

I am also grateful for the feedback I received from other colleagues of the University of Kashmir in particular Dr Javed Pervez and Mr. Sajad M. Khan of the P.G. Department of Computer Science. I thank all the people who have been part of my group particularly Er. Muheet Ahmad Butt and Er. Majid Zaman, Mohd Rafi Khan who would exchange their information regarding conferences and journals and give suggestions during our research publications and thesis preparation.

A good support system is important to surviving and staying sane in Post Graduate Department. I was lucky to have a strong supportive team, I thank the members of staff from P.G. Department of Computer Science and the Academic section for their help with administrative issues. Among many from whose help I benefited are Mr Mohd Shafi Mir, Mohd Younis wani, Mohd Ishaq khan and others.,I would like to thank those who shared their knowledge with me and enabled me to complete the work described in this thesis. I would like to express my deepest gratitude to Prof. Aarne Rante, University of Chalmers, for introducing me to the fascinating world of Natural Language Processing, Prof Chandershaker (BHU) for introducing me to different problems of NLP, the authors of papers and books cited in this thesis, to the University of Kashmir for making available an excellent environment for research, to my fellow-colleagues and other research scholars for our enriching discussions. I am also grateful to all the people who made my Ph.D. studies a pleasant experience and for being very supportive in many different ways.

Last but not least, I'd like to express my deepest gratitude to my father, mother and my wife, Gousia zahoor,for her patience and tolerance over the last four years. Gousia, I could not have been able to finish this work without your support. Thank you for being with me and for your appreciated sacrifices.

Manzoor Ahmad

Abstract

Spoken Language systems are going to have a tremendous impact in all the real world applications, be it healthcare enquiry, public transportation system or airline booking system maintaining the language ethnicity for interaction among users across the globe. These system have the capability of interacting with the user in different languages that the system supports. Normally when a person interacts with another person there are many non-verbal clues which guide the dialogue and all the utterances have a contextual relationship, which manage the dialogue as its mixed by the two speakers. Human Computer Interaction has a wide impact on the design of the applications and has become one of the emerging interest area of the researchers. All of us are witness to an explosive electronic revolution where lots of gadgets and gizmo's have surrounded us, advanced not only in power, design, applications but the ease of access or what we call user friendly interfaces are designed that we can easily use and control all the functionality of the devices. Since speech is one of the most intuitive form of interaction that humans use. It provides potential benefits such as hand-free access to machines, ergonomics and greater efficiency of interaction. Yet, speech-based interfaces design has been an expert job for a long time. Lot of research has been done in building real spoken Dialogue Systems which can interact with humans using voice interactions and help in performing various tasks as are done by humans. Last two decades have seen utmost advanced research in the automatic speech recognition, dialogue management, text to speech synthesis and Natural Language Processing for various applications which have shown positive results. This dissertation proposes to apply machine learning (ML) techniques to the problem of optimizing the dialogue management strategy selection in the Spoken Dialogue system prototype design. Although automatic speech recognition and system initiated dialogues where the system expects an answer in the form of ‘yes’ or ‘no’ have already been applied to Spoken Dialogue Systems(SDS), no real attempt to use those techniques in order to design a new system from scratch has been made. In this dissertation, we propose some novel ideas in order to achieve the goal of easing the design of Spoken Dialogue Systems and allow novices to have access to voice technologies. A framework for simulating and evaluating dialogues and learning optimal dialogue strategies in a controlled Natural Language is proposed. The simulation process is based on a probabilistic description of a dialogue and on the stochastic modelling of both artificial NLP modules composing a SDS and the user. This probabilistic model is based on a set of parameters that can be tuned from the prior knowledge from the discourse or learned from data. The evaluation is part of the simulation process and is based on objective measures provided by each module. Finally, the simulation environment is connected to a learning agent using the supplied evaluation metrics as an objective function in order to generate an optimal behaviour for the SDS.

CONTENTS

List of Figures xii

List of Tables xiv

List of Algorithms xvi

Glossary xvi

1 Introduction 1

1.1 Introduction 2

1.2 Motivation 3

1.3 Thesis Outline 4

1.4 Contributions 6

2 Spoken Dialogue Systems 7

2.1 History 9

2.2 What is a Human - Computer Dialogue ? 12

2.3 Levels in a Speech based Interaction 12

2.4 General Dialogue System 15

2.5 Spoken Dialogue System 15

2.6 Characteristics of spoken dialogue system 19

2.6.1 Turn-taking 19

2.6.2 Grounding 19

2.6.3 Conversational implication 20
2.7 Components of a Spoken Dialogue System 21
2.7.1 Speech Recognition 21
2.7.2 Natural Language Understanding 22
2.7.3 Dialogue Management 23
2.7.4 Speech Synthesis System(TTS) 24
2.8 User Simulation 25
2.9 Dialogue Manager Design Paradigms 26
2.9.1 Hand Crafted Approach 27
2.9.2 Sequential Decision Process approach 28
2.9.2.1 Markov Decision Processes framework(MDPF) 29
2.9.3 Partially Observable Markov Decision Process (POMDP) Framework 31
2.9.3.1 Process History 32
2.9.3.2 Performance Measures 33
2.9.3.3 Policy 33
2.10 Conclusion 34

3 Dialogue, Representation, Inference and Learning Strategies 35
3.1 Dialogue State Representation 37
3.1.1 Probabilistic Graphical Models 38
3.1.2 Bayesian Networks 38
3.1.2.1 Dependencies 40
3.1.2.2 History Nodes 41
3.1.2.3 User Acts and Observation Nodes 41
3.2 Factor Graphs 42
3.3 Loopy Belief Propagation Algorithm 44
3.3.1 Belief Propagation 44
3.4 Limiations 45
3.5 Expectation propagation 46
3.6 Comparison to previous work 48
3.7 Grouped Loopy Belief Propagation 49
3.8 Policy Design and Learning 49
3.8.1 Function approximation 50

3.9 Natural Actor Critic Algorithm 51
3.10 Evaluation 54
3.11 Conclusion 54

4 Language Models 56

4.1 Language Models 57
4.2 Historical background 58
4.3 Statistical Language Modelling 59
4.4 Statistics Language Modelling techniques 59
4.5 Class based Semantic n-gram Model 59
4.6 Proposed Model Description 60
4.7 Experiment 63
4.8 Result and Discussion 63

5 Semi-supervised Learning with Hidden State Vector Model 65

5.0.1 Machine Learning 66
5.1 The Hidden Vector State Model 67
5.2 Semi-Supervised Learning 69
5.3 Related Work 69
5.4 Proposed Framework 70
5.5 Distance-Weighted Nearest Neighbour Algorithm 72
5.6 Semi-supervised Learning based on expectation maximization 73
5.7 Experimental Evaluation 74
5.7.1 Methodology 74
5.7.2 Results 75
5.8 Conclusion 76

6 Prosodic Learning 77

6.1 Introduction 78
6.2 Corpus and Certainty annotation 80
6.3 Prosodic Model 81
6.4 Classification Results 82

7 Conclusions & Future Work 88
7.1 Conclusions Drawn 89
7.2 Future Work 90

References 92

Publications 101

LIST OF FIGURES

2.1 Architecture of a general Dialogue System 16
2.2 Architecture of a Spoken Dialogue System 18
2.3 Dialogue Act showing User simulator instead of User 25
2.4 Four major paradigms in spoken dialogue management 27
2.5 Casual Relationships between MDP states, actions, rewards 30
2.6 Influence Diagram in a POMDP framework 32

3.1 A portion of Bayesian network representing the POMDP Model 39
3.2 Factorisation of Bayesian network representing part of a health based information system 40
3.3 Bayesian network with splitted user-acts 42
3.4 Factor graph representing POMDP model 43
3.5 A Simple factor graph 47
3.6 Plot showing the trend in the reward during policy training with a sample 100 dialogues 54

4.1 Semantic Parsing Example 61

5.1 Semantic Parsing Example 76

6.1 Undiscretized Data Decision Tree 83
6.2 Decision tree obtained after data discretisation 84
6.3 Figure indicating the Pitch versus the output of different samples . . . 85

6.4 Figure indicating the maximum intensity versus the maximum pitch of different samples 86
6.5 Performance Statistics 87

LIST OF TABLES

3.1 Example split of overall observation into concept level observations . . 41

4.1 Perplexity of Static and Adaptive Model. 63
4.2 Influence of Cache Size on perplexity of TRAINS Corpus 64

5.1 Precision %age of Semi-Supervised Learning on Labeled and using labeled training data for unlabeled data set 75

6.1 Percentage of corpus with different levels of certainty, annotated by listening to the audio of the dialogue context and annotated based on the lexical structure of the dialogues. 81
6.2 Extracted and selected features. 82

LIST OF ALGORITHMS

3.1 Loopy Belief Propagation Algorithm 45
3.2 Expectation Propagation Algorithm 48
3.3 Natural Actor Critic Algorithm 53
5.1 Minimum Edit Distance Algorithm 72

GLOSSARY

ASR Automatic Speech Recognition ; can be defined as the independent, computer driven transcription of spoken language into readable text in real time

DT Discriminative training ; is based on comparison the likelihood scores estimated for single speech units(phones, words).

GM Generative model ; are a class of models for randomly generating observable data, typically given some hidden parameters. It specifies a joint probability distribution over observation and label sequences

HMM Hidden Markov Model; is a statistical tool for modelling a wide range of time series data, a model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states

MMI Maximum Mutual Information ; considers HMMs of all the classes simultaneously during training. Parameters of the correct model are updated to enhance it's contribution to the observations, while parameters of the alternative models are updated to reduce their contributions. This procedure belongs to the "discriminative training" category

PP Perplexity; is often used for measuring the usefulness of a language model and is a probability distribution over sentence, phrases, sequence of words, etc .

SDS Spoken Dialogue System; a software agent that interacts with humans by accepting spoken language as input

Stationary Policies Policies which do not depend on the stages

Turn A period in which one of the participants in a dialogue has a chance to say something to the other participants

User Simulator A simulation of dialogue system users, which generates a dialogue act given a dialogue history.

WAcc Word Accuracy; used to report the performance of a speech recognition system

$$\begin{aligned} WAcc &= 1 - WER \\ &= \frac{N - S - D - I}{N} \\ &= \frac{H - I}{N} \end{aligned}$$

where
H is $N - (S + D)$, the number of correctly recognised words

WER Word Error Rate; A common metric of the performance of a speech recognition or machine translation system.

$$WER = \frac{S + D + I}{N}$$

or

$$WER = \frac{S + D + I}{S + D + C}$$

where

- S is the number of substitutions,
- D is the number of deletions,
- I is the number of insertions,
- C is the number of the corrects,
- N is the number of words in the reference ($N = S + D + C$)

Word-lattice A structure for representing multiple speech-recognition hypotheses.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Spoken language is the most intuitive form of communication between humans. Computer can be added knowledge and made to act intelligently but they lack this property to communicate in natural spoken language. If the computers are made to interact with the humans in a natural way using spoken language, it would greatly simply their usage which otherwise have obstructed many humans in their usage. The study of these systems, known as spoken dialogue systems, is an important area of engineering and provides insight into the understanding of human learning, linguistics and artificial intelligence. Early research on these systems was done about text interfaces rather than speech which produced remarkable results. Reasonably convincing examples of text-based dialogue systems were soon built using relatively simple techniques. The ELIZA program was one example, which attempted to impersonate a psychiatrist (Weizenbaum, 1966). The program reordered words in an input sentence using simple pattern-matching rules. A sentence like 'I am feeling sad.' would be converted into 'Why are you feeling sad' without the program truly analyzing the meaning of the original input. This simple scheme was so effective that when Weizenbaum gave it to his secretary, she thought the machine was a real therapist and spent hours revealing her personal problems to the program (Wallace, 1995; Weizenbaum, 1976). She was then horrified to find out that all dialogues with the program had been recorded and that her boss had access to all the transcripts. As research continued it became apparent that these simple schemes were difficult to extend to more meaningful tasks. The state-of-the-art is so far away from building a general purpose dialogue system that Marvin Minsky, known by some as the father of artificial intelligence, even suggested that research on this topic would be 'obnoxious and stupid' (Sundman, 2003). On the other hand, dialogue systems where the topic of conversation is limited have been shown to have wide application and have improved significantly in performance over recent years. This thesis will deal exclusively with these learning dialogue strategy in limited-domain dialogue systems. Examples of such dialogue systems are numerous. Systems have been deployed to provide train information, in-car navigation, make bookings, interact with robots and build computer interfaces for the illiterate and blind.

The rest of this chapter is organized as follows: Section 1.2 explains the motivation behind this work, Section 1.3 gives an outline of this thesis followed by Section 1.4 which presents contributions made in this work.

1.2 Motivation

Computer-based dialogue systems can be used to replace humans for such tasks where the need is 24x7 such as enquiry system in hospitals, ticket reservation systems, aids for illiterate and assistive technology based devices. This is based on several key advantages of such artificially intelligent systems i) Such devices will be available at all times ii) These devices can be customized as per individual tastes and interests iii) They will never get tired iv) Cost effective v) allows for more user privacy vi) can be used in situations where it is impossible for humans to be placed. Existing spoken dialogue systems are far from perfect due to various limitations and three major faults appear when one compares them to humans.

1. Humans are capable of holding a dialogue in significantly more difficult environments and are not as affected by noise, ambiguities and errors.

2. Humans can learn during the conversation and do not need explicit rules to predefine how they should interact.

3. Humans are not constrained with respect to one domain but posses the capability to understand any topic of conversation and can even turn between multiple conversations.

4. Humans use other para linguistic information in the form of visual cues and prosodic information to phrase its response while interaction.

In order to effectively handle noise, ambiguities and other difficulties, this thesis will argue that a adaptive hybrid language model must be used for handling uncertainty in the dialogue during the speech recognition. Spoken language is an inherently error-filled medium of communication and even humans use error-correcting strategies to ensure a correct understanding. n-gram Language model is a appealing approach for decision making and there is also evidence that humans use language models when they have to predict the words which are either in low tone or not clear. Bayesian probability theory is therefore an appropriate framework for handling the resulting uncertainty. This thesis will argue further that statistical methods are an appropriate framework for allowing a system to learn what to say in a dialogue. By examining the effects of past decisions, a system can learn statistically optimal rules for making future decisions. The resulting systems are capable of adapting to new environments, new types of user and new domains with limited additional development. A significant emphasis will be placed on building systems for large domains. The task of building a general purpose system is not covered, although it is possible that suitable learning methods may enable future general purpose systems to be built. An approach which requires developers to encode all knowledge about the world is unlikely ever to succeed. An approach which allows the system to learn this knowledge for itself may one day become a reality.

1.3 Thesis Outline

Spoken dialogue systems are made up of several complex components. This thesis will be concerned largely with the decision making component, known as the dialogue manager. In studying the dialogue manager, it is useful to understand how each of the other components interact to form a complete system.

Chapter 2 will begin the main part of this thesis with a discussion of a Spoken Dialogue System and its various components. Dialogue management which is the heart of a spoken dialogue system is also elaborated and various error handling mechanisms are also discussed. Chapter 2 also provides a literature review of dialogue management theory. The task of dialogue management can be separated into two main areas. Firstly, the system must efficiently maintain the state of the dialogue and secondly, the system must use the state to make appropriate decisions.

Chapters 3 discuss methods for maintaining this state and discusses the actual decision making. (Robert, 2007, Ch 11) gives some evidence that the Bayesian approach is in fact the only coherent approach to decision making. Various other theoretical reasons for Bayesian decision making are also given. Lewandowsky et al. (2009) and Wolpert & Ghahramani (2005) provide example experiments illustrating the Bayesian nature of human decision making.

Chapter 3 shows how machines can automatically decide what to say in a dialogue. The chapter suggests the use of the Dynamic Bayesian Networks for representation of the spoken dialogue and Grouped Loopy Belief Propagation algorithm for inferencing the state representation and Natural Actor Critic algorithm for optimisation of the dialogue policy and provides a framework for building suitable function approximations in the case of dialogue. An example application in the Tourist Information domain is also described

Chapter 4 proposes an adaptive cache POS based language model which has been used to cope up with the uncertainty which may be because of the spoken communication limitations or even when the user himself is not clear about which word to use to fill in the gap.The proposed framework based on dynamic probabilistic model uses word dependencies based on their part of speech tags along with the tri-gram Model but also takes care of the influence of the word which are very far from the word being considered in a text and stores the word history in a dynamic cache for information mining using long distance dependency.

Chapter 5 proposes a framework wherein standard algorithms from the machine learning literature have been used to efficiently annotate the un-annotated utterances using hidden vector state model. The algorithm, called Mean Weighted Edit Distance and Maximum Expectation are implemented which, improves efficiency by exploiting conditional independence assumptions between variables in the probabilistic model. The chapter provides an introduction to Hidden Vector State Model, Semi Supervised Learning, Mean Weighted Edit Distance and Maximum Expectation The chapter also provides an example of the use of this proposed framework.

Chapter 6 shows how prosodic information of the user utterance can be used to know his mental state. This chapter discusses about how different parameters like pitch, intensity, utterance time are used to determine the users belief about a particular concept and his confidence levels during the dialogue. Results of Two experiment one in the area of tourism and other about a university lecture are depicted to indicate the level of certainty of different users.

Chapter 7 summarizes the major findings of this work and the contribution to knowledge made in this dissertation. Moreover, it also presents the future scope of this work which can be investigated in further research.

Some of the material presented in this thesis has been published previously. The complete list of these published articles immediately follows Chapter 6.

1.4 Contributions

In this thesis, we have mainly concentrated on the framework for the dialogue manager module of the Spoken Dialogue System. The main objective was to make a automatic policy/strategy formulation by dialogue manager based on the previous dialogue history into consideration. The major contributions are listed as follows:

1. We surveyed existing representation for the dialogue states and algorithms that existed for updating the beliefs in the environment states. The goal was to identify the optimum representation and algorithm which could enable efficient inferencing. we proposed Dynamic Bayesian network for the dialogue states and Group Belief Propagation algorithm for inferencing which proved to be computationally efficient. The Natural Critic Algorithm which is a modified version of gradient descent has been used to learn the policy and has enabled to learn the dialogue process optimally using the given structures.

2. We empirically evaluated various techniques for improving the user utterance recognition to improve over the confidence score of the automatic speech recognition.We proposed adaptive language model and semi-supervised learning over Hidden Vector state language model for this purpose

3. Lastly, we proposed that the prosodic information variables, like the frequency, pitch, intensity, etc. can be utilized to improve the dialogue policy formulation, Experiments were conducted which proved that this information can aid the the dialogue manager in having an meaningful conversation with the user.

CHAPTER 2

SPOKEN DIALOGUE SYSTEMS

Thoughts and ideas are meaningful and can be realized when they can communicated to others and spoken language is one of the human being's main characteristic by which he can communicate his ideas to other and collaboratively realize them in practice other than in the written form of language. The speed and ease with which we can speak is comparatively more than any other form of interaction like key presses or mouse movements. But in real life when we speak we unknowingly embed lot of information in the form of pitch, intensity, temporal variation other than the linguistic word lattices which build our utterances during a dialogue with other human. And humans being an intelligent entities decode not only the linguistic information but also the auditory information and uses the previous dialogue context to generate a response or decide about the next move in the dialogue process. The ability of understanding and producing more or less coherent answers to spoken utterances implicitly defines different degrees of intelligence which most humans demonstrate in all situations. But intelligence has a closely relationship with learning.

Intelligent systems are agents which are capable of acting rationally towards any change in environment based on the valid information it has. Machines like computers can be intelligent if they are able to act rationally towards any change in environment by applying the knowledge it carries in its knowledge-base. Since humans use mainly spoken language and natural language for interaction with other human, machine to be human-like have to also use the spoken language as its means of input and output for interaction. There are many issues in the design of a spoken dialogue system. and two key assumptions are almost always taken.

i Dialogues with exactly two participants are considered

ii All interactions between the system and the user are in the form of turns. A turn in a dialogue is a period in which one of the participants has a chance to say something to the other participant.

Under these assumptions, the dialogue will follow a cycle, known as the dialogue cycle. One of the participants says something, this is interpreted by the listener, who makes a decision about how to respond and the response is conveyed back to the original participant. This same process of listening, understanding, deciding and responding then occurs on the other side and the cycle repeats. Hence any dialogue system requires a number of components: one that can understand the user‘s speech, one that makes decisions and one that produces the system‘s speech.

2.1 History

Machines producing speech is not new. Since 17th century mathematicians and logicians who designed the first computational machines had the thought that machines could speak which was clear from the Rene Descartes declaration from “ Discourse on the method” that if machines bearing the image of our bodies and capable of imitating our actions, we may easily conceive a machine which emits vocables and even acts in response to change in its organs so as to reply what is said in its presence. Then Lenord Euler in 1761 said that,“It would be a considerable invention indeed that of a machine able to mimic speech with its sounds and articulations. I think its is not impossible”. In 1779 Christain Kratzenstein designed a machine which was able to produce vowel sound and was based on the human vocal tract. In 1791, Wolfgang Von Kempelen built the machine also known as first talking machine which was not only able to produce sounds but also words and even short sentences in Latin, Italian and French languages. At the end of 1878 Alexander Graham Bell invented telephone which was based on his inspiration to invent a machine that could transcribe spoken words into text which he could not complete.

While AT&T Bell Laboratories developed a primitive device that could recognize speech in the 1940s, researchers knew that the widespread use of speech recognition and understanding would depend on the ability to accurately and consistently perceive subtle and complex verbal input. In 1939 Homer Dudley of Bell Labs invented a controlled speech synthesizer but it required highly trained technicians to use it. In 1942 a toy dog which responded to its name was produced by Elmwood Button Company that created a landmark in the field of speech recognition. Since then lot of research in the parallel and inter-disciplinary fields have contributed to this area of speech analysis and synthesis in building machines which could respond to speech signal as a mean of interaction. In 1952, Bell Labs developed a system which recognized spoken digits transmitted by a phone with an accuracy of 98 % with speaker adaptation [Davis et al., 1952]. In 1959, a speaker independent system able to recognize vowels with an accuracy of 93% was developed by Forgie and Forgie at MIT. A system capable of matching spoken utterance to a list of 50 words with an accuracy of 83 % along with a confidence score to indicate the recognition result was developed by Ben Gold of MIT in 1966. In 1956, Noam Chomsky developed many theories about linguistics and computational grammars and built the foundations of Natural Language Processesing. In 1950 Claude E. Shannon, also known as father of information theory used the concept of artificial intelligence to develop a chess playing software by a machine. Thus, in the 1960s, researchers turned their focus towards a series of smaller goals that would aid in developing the larger speech recognition system. As a first step, developers created a device that would use discrete speech, verbal stimuli punctuated by small pauses In 1966 Joseph Weizenbaum from MIT developed ELIZA, the first artificial intelligence program which simulated human conversation and passed Turing Test to prove the intelligence quotient. In 1968, Arthur C. Clarke and Stanley Kubrick created HAL9000 a computer that could hold a conversation, think and adapt its behavior.

However, in the 1970s, continuous speech recognition, which does not require the user to pause between words, began. Also lot of research was funded in the speech understanding program. It aimed at analyzing, storing and understanding continuous speech by the computer systems. This led to lot of research groups at many leading universities like MIT, Stanford, Carnegie Mellon University and other research institutions like Microsoft and IBM to ponder on the different issues of speech understanding. More focus in these days was on Automatic Speech recognition, where the recognition error rates high because of the smaller vocabularies, It was because of the statistical and empirical pattern matching frameworks based on Hidden Markov Models used by James, Janet Baker and Fedreick Jelinek who actually got a break though in the area of statistical pattern matching framework based on Hidden Markov Models to speech recognition [Jelinek, 1976]. In the same years, due to technology improvement especially memory and Processesing power of the computers, the structure of human discourse was the main theme for investigation by researchers for making human computer interfaces more friendly [Rabiner and Schafer, 1978]. In 1974 Barbara Grosz studied the structure of dialogues in collaborative tasks [Grosz, 1974]. Speech Recognition Systems have become so advanced and mainstream that business and health care professionals are turning to speech recognition solutions for everything from providing telephone support to writing medical reports. Technological advances have made speech based systems and devices more functional and user friendly, with most contemporary products performing tasks with over 90 percent accuracy. In 1986, Barbara Grosz and Candace Sidner developed the theory of centering [Grosz and Sidner, 1986] that aimed to formalize the way a human follows the focus of a conversation. James

Allen applied statistical pattern matching techniques usually applied in speech recognition to semantic parsing of natural language [Allen, 1987]. In 1990's hybrid methods combining Artificial neural networks and HMM's were successfully used in large speech recognition systems. [Bourlard and Morgan, 1994] In 1996, the development of a complete spoken dialogue systems (SDS) which including automatic speech recognition, Natural Language understanding, dialogue management and speech synthesis started to emerge.

Today, the latest generation of speech technology delivers conceptual search. This approach utilizes advanced mathematics and complex algorithms to derive meaning from speech. Conceptual search addresses the shortcomings of previous speech technology models and provides the most accurate way of recognizing and finding speech because it understands what is being said. It can distinguish between homophones, heteronyms, as well as find and group things by concept. It can also find related information based on meaning and has lower computational need than some of the earlier generations of speech recognition technology According to the industry, Satisfying the needs of consumers and businesses by simplifying customer interaction, increasing efficiency, and reducing operating costs, speech based software is used in a wide range of applications. Indeed, recent advances in spoken dialogue systems are creating a dynamic environment, since this technology appeals to anyone who needs or wants a hands-free approach to computing tasks. As the merger of large vocabularies and continuous recognition continues, look for more and more research is taking place toward speech based systems and researchers are developing new gadgets with this technology.

Today, the latest generation of speech technology delivers conceptual search. This approach utilizes advanced mathematics and complex algorithms to derive meaning from speech. Conceptual search addresses the shortcomings of previous speech technology models and provides the most accurate way of recognizing and finding speech because it understands what is being said. It can distinguish between homophones, heteronyms, as well as find and group things by concept. It can also find related information based on meaning and has lower computational need than some of the earlier generations of speech recognition technology According to the industry, Satisfying the needs of consumers and businesses by simplifying customer interaction, increasing efficiency, and reducing operating costs, speech based software is used in a wide range of applications. Indeed, recent advances in spoken dialogue systems are creating a

dynamic environment, since this technology appeals to anyone who needs or wants a hands-free approach to computing tasks. As the merger of large vocabularies and continuous recognition continues, look for more and more research is taking place toward speech based systems and researchers are developing new gadgets with this technology.

2.2 What is a Human - Computer Dialogue ?

Dialogue may be defined as an interaction / a spoken or written conversation exchange between two agents based on a sequential turn taking with an aim of achieving some goal. When one of the agent is a computer and the other is human, the dialogue is known as Human- Computer Dialogue. Also When the system initiates the dialogue and always prompts the user to select an utterance from fixed menus it is known as system initiative dialogue system. When the human and the machine makes a more natural dialogue where the system attempts to determine the intentions of the user from the unrestricted utterances, the dialogue system is known as mixed initiative dialogue system. But when other mean of communication like facial expressions, prosodic information etc. other than speech is used in the interaction its known as multi-modal dialogue. When the human machine dialogue is dedicated to the realization of a particular task or set of tasks, the dialogue system is known as task oriented dialogue system. When the agent in a dialogue is a spoken dialogue system, the user and the system exchanges a series or utterances where each spoken utterance is the acoustic realization of the intentions and concepts embedded in the form of word lattice that one of the agents wants to communicate to the other. Human-to-computer interaction is an form of natural language Processing task between human and the computer where the elements of human language, be it spoken or written, are formalized so that a computer can perform value-adding tasks based on that interaction.

2.3 Levels in a Speech based Interaction

Information conveyed by speech can be analyzed at several levels. In the field of Natural Language Processing, seven levels are commonly admitted in order to describe speech-based communication [Boite et al., 2000]. These levels can be classified into high and low levels of description, the lower level starts from the physical sound signal. This distinction between high and low levels is applicable to all types of communications as there is always a possibility to distinguish the physical stimuli and the interpretation.

(1) **The Acoustic Level**

Speech is a sequence of sounds which may also be defined as a variation of the air pressure created by the vocal tract. The acoustic level concerns the signal and as such represents the lowest level of speech communication. The study of the acoustic signal includes the study of any of its representation as the electrical output of a microphone (analog or digital), wave forms, frequency analysis (Fourier transforms, spectrograms) etc. Useful information can be obtained from the analysis of the acoustic signal such as the pitch (fundamental frequency), the energy and the spectrum. In general, it is the only level considered by speech coding techniques. As the human vocal tract is a physical instrument, it is subject to a certain inertia and thus, it cannot assume sudden state modifications. This results in an important property it can be considered as a pseudo-stationary signal.

(2) **The Phonetic Level**

It is a low level description where the main focus is on the production of particular sounds by the ariculatory system. The phonetics studies how humans voluntary contracts muscles in order to dispose obstacles like tongue, lips, teeth and other organs in the aim of pronouncing a specific sound.

(3) **The Prosodic Level**

The main task at this level is the analysis of a limited number of distinct sounds allowed in a particular language (phonemes), the rhythm with which they are produced in a sequence, the musicality applied to this sequence(prosody) and he accentuated part within the sequence. This level is considered to be transitory between low and high levels as it concerns physically observable features of the signal but those specific traits are voluntary produced by the speaker in the aim of including meaningful clues into the speech signal. Prosody is used to detect the sentiments and emotions in the speech signals like tutoring applications [Litman and Forbes, 2003].

(4) **The Lexical Level**

Also known as morphological level,The main focus at this level is on all the valid phoneme sequences that produce words included in the lexicon of a particular language where each phonemes are the finite number of different sounds in a specified language. This level forms the first stage for the high level where word elementary sub-units are studied which convey sense.

(5) **The syntactic Level**

Words constitute a valid sentence in a language only when the word chain follows the set of rules also known as the grammar of the language or syntax of the language. There are different rule sets to describe the syntax. The main function of the grammar is to make a word function in a sentence so that the sentence follows the syntactic structure. Computational grammars which are different from the linguistic grammars have been developed in the early ages of natural language understanding [Chomsky, 1965].

(6) **The Sematic Level**

At this level, the main focus is to determine the context independent meaning that the words in the sentence mean and how those meanings combine to form the information that the sentence try to convey. Although an utterance may be syntactically correct but it might not provide the coherent information for which it was framed. So this level studies how to extract the meaning/sense from the utterances.

(7) **The Pragmatic Level**

Pragmatics is the study of grouping all the context dependent information in a dialogue. Most often the utterances implicitly refer to the underlying information also known as ground information. which is expected to be known by the participants of the conversation either based on the environmental conditions, the beliefs that the participants hold, their background, common knowledge that they hold. Pragmatic level is divided into three sublevels [Allen et al., 1987].

Pure pragmatic level

the study of the different meanings that can convey a single sentence uttered in different contexts.

Discourse Level

concerns how the directly preceding sentence affects the interpretation of the next sentence. The study at this level help in disambiguation and anaphora resolution.

World Knowledge Level

Also known as ground knowledge includes all the information people know about the world and what an participant in conversation knows about the other participants belief and goals.

2.4 General Dialogue System

In human to human conversation, the conversant tries to integrate all the information from the senses based on the knowledge

2.5 Spoken Dialogue System

Spoken dialogue interaction has been suggested by researchers and practitioners as a promising alternative way of communication between humans and machines[Zue et al., 2000]. A compelling motivation is the fact that conversational speech is the most natural, efficient, and flexible means of communication among human beings. Because of the complexity of human-human interaction, human-machine conversations need to be much simpler. Talking to a machine requires a spoken dialogue system. These systems may be alternatively referred to in the literature as "conversational agents", "Spoken language systems" or "conversational interfaces" [Jurafsky and Martin, 2008], [McTear, 2004] Huang et al., 2001. Such systems should be able to understand what a person says, take an appropriate action, and then provide a response. Ideally, spoken dialogue systems should yield successful, efficient and natural conversations within a given domain. However, building such systems is still a challenge for science and engineering. Thus a spoken dialogue system may be defined as a intelligent agent that interacts with humans using spoken language in order to perform some task which is normally to access and manage information. These systems are an example of an open ended, goal oriented, real time interactions between humans and computers. A Multi Modal Spoken Dialogue systems is one which uses many modes of input like speech, Graphic User

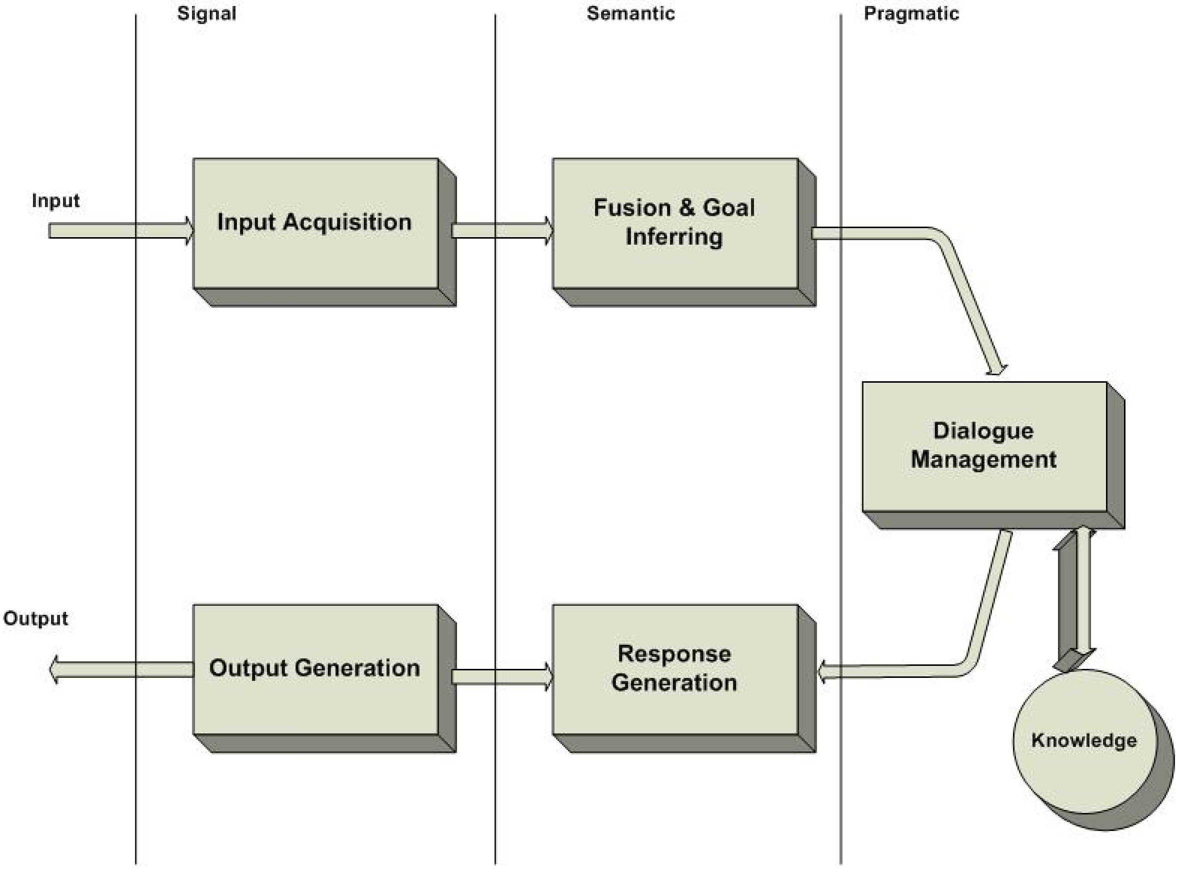


Figure 2.1: Architecture of a general Dialogue System - The figure shows different processes involved in a human to human interaction

Interface and computer vision e.g. In case of Telephonic technical support for product and services, In-car music control for music navigation, Tutoring, Language learning, Mobile search interface, Computer based assistance technology especially in Eldercare, Automated receptionist. Voice enabled interfaces are now becoming common and most of us have used such interface while dialing the number of a contact using his speech tag, using voice recognition software for typing our documents in a word Processesing software, browsing the internet using voice enabled internet browsing software which accepts our voice commands and hear the emails in the inbox along with their contents. In general the classification of spoken dialogue systems depends on the application and its complexity and are becoming ubiquitous due to their rapid improvement in performance and decrease in cost. The spoken dialog systems receive speech inputs from the user, and the system responds with the required action and the information. For example, a user might use a spoken dialog system to reserve a flight over the phone, to direct a robot to guide him to a specific room, or to control in-car devices such as a music player or a navigator. Since the early 1990s, many spoken dialog systems have been developed in the commercial domain to support a variety of applications in telephone-based services. For example, early spoken dialog systems functioned in restricted domains such as telephone-based call routing systems (HMIHY) [Gorin et al. 1997], weather information systems (JUPITER) [Zue et al. 2000], and travel planning (DARPA communicator) [Walker et al. 2001]. More recently developed systems are used in incar navigation, entertainment, and communications [Minker et al. 2004; Lemon et al. 2006; Weng et al. 2006]. For example, the EU project TALK2 focused on the development of new technologies for adaptive dialog systems using speech, graphics, or a combination of the two in the car. More recently, multi-domain dialog systems have been employed in real life situations [Allen et al. 2000; Larsson and Ericsson 2002; Lemon et al. 2002; Pakucs 2003; Komatani et al. 2006]. Such multi-domain dialog systems are now able to provide services for telematics, smart home, or intelligent robots. These systems have gradually become capable of supporting multiple tasks and of accessing information from a broad variety of sources and services.

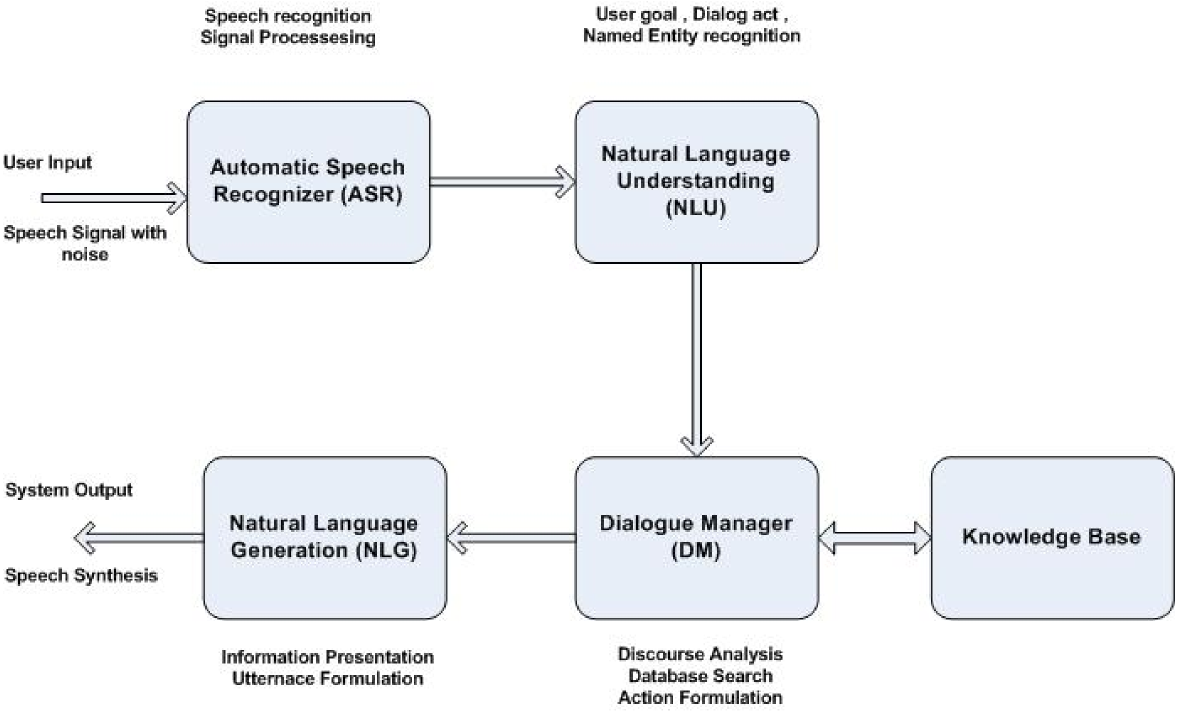


Figure 2.2: Architecture of a Spoken Dialogue System - The figure shows different modules involved in a human to computer interaction

2.6 Characteristics of spoken dialogue system

2.6.1 Turn-taking

A dialogue consists of many turns, where in every turn one participant speaks. Who should speak in the next turn is determined by using the Turn-Taking rules at the end of each turn. These rules apply at a *Transition-Relevance place(TRP).* Transition relevance places where the structure of the language allows speaker shift to occur. Here is a simplified version of the turn-taking rules, grouped into a single three-part rule.At each TRP of each turn:

1. If during this turn the current speaker has selected A as the next speaker then A must speak next.
2. If the current speaker does not select the next speaker, any other speaker may take the next turn.
3. If no one else takes the next turn, the current speaker may take the next turn.

2.6.2 Grounding

Grounding is a collective act performed by the speaker and the hearer wherein both must constantly establish common ground. By common ground we mean the set of things that are mutually believed by both speakers. The need to achieve common ground means that the hearer must ground or acknowledge the speaker's utterances, or else make it clear that there was a problem in reaching common ground. For example
U:I want to fly from Boston.
S: Mm hmm.(Acknowledgement)
The word mm-hmm here is a continuer, also often called a backchannel or an acknowledgement token. a continuer is a short utterance which acknowledges the previous utterance in some way, often cueing the other speaker to continue talking (Jefferson,1984;Schegloff, 1982;Yngve,1970).Clark and Schaefer(1989) discussed five main grounding methods, ordered from weakest to strongest:

- **Continued attention:** B shows she is continuing to attend and therefore remains satisfied with A's presentation. Relevant next contribution: B starts in on the next relevant contribution.
- **Acknowledgement:** B nods or says a continuer like "uh-huh", "yeah" an assessment like "that's great".
- **Demonstration:** B demonstrates all or part of what she has understood A to mean, for example by reformulating A's utterance or by collaborative completion of A's utterance.
- **Display:** B displays verbatim all or part of A's presentation.

Some Grounding examples:
U: ...I need to travel in June.
S: And, what day in May did you want to travel? (Display)
U: I want to be in New Delhi in time for the Asian games.
S: <reject>. Sorry, I didn't get that. Please say the month and day you would like to leave.(Rejection)
U: I want to go on October fifteenth.

2.6.3 Conversational implication

The interpretation of an utterance relies on more than just the literal meaning of the sentences. For example:
S: ...And, what day in May did you want to travel?
U: OK uh I need to be there for a meeting that's from the 12th to the 15th.
Here the user does not in fact answer the question. The user merely states that he has a meeting at a certain time. In this case the speaker seems to expect the hearer to draw certain inferences; in other words the speaker is communicating more information than seems to be present in the uttered words. Grice proposed that what enables hearers to draw these inferences is that conversation is guided by a set of maxims, general heuristics which play a guiding role in the interpretation of conversational utterances. He proposed the following four maxims:

- Maxim of Quantity: Be exactly as informative as is required:
  a. Make your contribution as informative as is required
  b. Do not make your contribution more informative than A required.

- Maxim of Quality: Try to make your contribution one that is true:
  a. Do not say what you believe to be false.
  b. Do not say that for which you lack adequate evidence.

- Maxim of relevance: Be relevant

- Maxim of Manner: Be clear, brief and orderly

2.7 Components of a Spoken Dialogue System

The general spoken dialogue system integrates four main components to process the speech signal from the user in presence of environmental noise and the system can generate the output which can be either visualized on the screen or synthesized by a text to speech synthesis module or a pre-recorded audio. This process works iteratively to complete the dialogue process wherein the intended purpose of the user is achieved. The components involved in the dialogue process are :-

2.7.1 Speech Recognition

The users makes a verbal response which is usually speech signals with noises which are recognized by an automatic speech recognition(ASR) subsystem which transforms the speech waveform into a sequence of parameter vectors which are then converted into a sequence of word (text). Most of the speech recognition methods uses Hidden Markov Model (HMM) to estimate the most probable sequence of words from a given speech signals. This component is built using many available toolkits ATK/HTK [Young et al., 2000] and SPHINX packages [Walker et al., 2004]. The performance of the speech recognition engine will depend on the difficulty of the task and on the amount of in-domain training data. The error rates are higher in the limited-domain dialogue systems and the user speaks freely using words which are out of the bounds of the list. [Raux et al., 2006] describes the Let's Go! bus information system, which has a sentence average word error rate of 64%. The word error rate of the ITSPOKE an intelligent tutorial system is 34.3% [Litman and Silliman, 2004].

Most current spoken dialogue systems use only the most likely hypothesis of the user's speech. State-of-the-art recognisers can however, output a list of hypotheses along with associated confidence scores. This list is called an N-best list, where N denotes the number of hypotheses. The confidence scores give an indication of the likelihood that the recogniser attaches to each word sequence. Ideally these confidence scores will give the posterior probability of the word sequence given the audio input [Jiang, 2005]. In some cases the recogniser may also return a word-lattice to represent the set of possible hypotheses. Such word lattices may be converted into N-best lists by first converting the lattice to a confusion network and then using dynamic programming to find the minimum sum of word-level posteriors [Evermann and Woodland, 2000].

2.7.2 Natural Language Understanding

This unit analyses the textual form for the set of hypothesis of the user utterance to understand the meaning of these words with the main aim to determine what the user wants to achieve by saying the words e.g morphological analysis, part-of-speech tagging, and shallow parsing. The NLU module maps the pre-processed utterance to a meaning representation (e.g., semantic frame) from which the dialogue act, user goal, and named entities are extracted by semantic parser. e.g, whether the user says "I'd like to know the doctor in the orthopaedics."or "who is the orthopaedician on duty " the desired outcome is the same. The user is asking for about the doctor on duty in orthopaedics department. The fact that the first utterance is a statement and the second is a question is irrelevant. This distinction between the exact semantics of an utterance and it's purpose was first made explicit in the definition of a speech act, which is a representation of this underlying action [Austin, 1962], [Searle, 1969]. In the example above, the speech act for both utterances would be "request".

The speech acts has been extended in the case of dialogue to include actions relating to turn-taking, social conventions and grounding [Traum, 1999]. The resulting concept is called a *dialogue act tag*. Dialogue act tags also allow actions such as "confirm" and "affirm" for confirmations and affirmations.e.g, a "confirm" action might be used to represent "Did you say you wanted to see an orthopaedician" and an "affirm" act might be used to represent "Yes!". In the traditional definitions of both speech and dialogue acts, the semantic information is completely separated from the act. A simplified form of semantic information is clearly an important input to the dialogue system. In the case of the user asking for the doctor on duty the information should be represented to indicate what is being requested. It is necessary to represent the dialogue act as “request(doctor)”. Similarly, the confirmation case above the dialogue act may be represented as “confirm(doctor=orthopaedic)”. Mostly the Dialogue acts are therefore represented as the combination of the dialogue act type followed by a (possibly empty) sequence of dialogue act items.

$$dialog\_act\_type(a = x, b = y, ...)$$

The *dialog_act_type* denotes the type of dialogue act while the act items, $a = x, b = y, etc.$will be either attribute-value pairs such as doctor=orthopaedic or simply an attribute name or value e.g request(addr), meaning “What is the address?” and inform(well), meaning “I am well”.With the concept of dialogue acts in hand, the task of understanding the user becomes one of deciphering dialogue acts. This is known as semantic decoding. In general one could imagine doing this on the basis of several sensory inputs. The prosodic information such as pitch or intensity of a user’s utterance might give some indication as to the dialogue act type.

There are a wide range of techniques available for semantic decoding. Hand-crafted techniques which include template matching and grammar based methods. Data-driven approaches include the Hidden Vector State model [He and Young, 2006], machine translation techniques [Wong and Mooney, 2007], Combinatory Categorial Grammars [Zettlemoyer and Collins, 2007], Support Vector Machines [Mairesse et al., 2009] and Weighted Finite State Transducers [Jurcıcek et al., 2009]. Most semantic decoders will assign exactly one dialogue act for each possible word sequence obtained from the speech recogniser. In the case of ambiguities, however, the semantic decoder may choose to output a list of the most probable outputs along with associated confidence scores. Since the speech recogniser is producing an N-best list of word sequences, some method must be found for combining the confidence scores from the speech recogniser with those of the semantic decoder.

2.7.3 Dialogue Management

After the utterances are semantically decoded, the system must choose an appropriate response from a set of alternatives based on some strategy. The component which makes these decisions is called the dialogue manager. The response chosen by the system is encoded as a dialogue act and is known as the system action or system act. The chosen response is selected from a set of possible actions, $a \in A$ and will depend on the input that the system receives from the semantic decoder. This input is called the observation, labelled $o \in O$, since it encodes everything that the system observes about the user. Choosing the best action requires more knowledge than simply the last observation. The dialogue manager coordinates the activity of all components, controls the dialogue flow, and communicates with external applications. The dialogue manager should play many roles which include discourse analysis, knowledge database query, and system action prediction based on the discourse context and dialogue history which plays an important role. The dialogue manager takes this into consideration by maintaining an internal representation of the full observation sequence. This is called the dialogue state, system state or belief state and is denoted by $b \in B$. The current belief state will depend on a belief state transition function which is a mapping $\delta : A \times O \times B \to B$ which takes a given belief state and updates it for each new observation and system action.

The component of the dialogue manager which defines its behaviour is the dialogue policy or dialogue strategy(π). The policy determines what the system should do in each belief state. In general, the policy will define a probability distribution over which actions might be taken. If $\pi(A)$ denotes the set of these distributions then the dialogue policy will be a mapping from belief states to this set, $\pi : B \to \pi(A)$. Clearly the actions, belief states and observations are all indexed by the turn number. When it is important to note the time step being considered, they are denoted a_t, b_t and o_t. While the system is in state b_t it will choose action a_t according to the distribution determined by the policy, $\pi(b_t)$. The system then observes observation o_{t+1} and transitions to a new system belief state b_{t+1}. When exact point in time is insignificant, the t is omitted and a prime symbol is used to denote the next time step (e.g. $o = o_{t+1}$).

2.7.4 Speech Synthesis System(TTS)

The system responses have to be finally conveyed to the user. The system dialogue acts are first converted to natural language with a list of content items from a part of the the knowledge base that keeps track of all the information generated through the dialogue history, which is queried and/or updated by the natural language generator and finally passed to Speech Synthesis system which conveys the message as audio. The simplest approach for natural language generation is to use templates. As an example, a template might transform “inform(doctor=x)” into “The doctor is x”, where “x” may be replaced by any name of the doctor which will be queried from the database. Templates have proven to be relatively effective for natural language generation, since the number of system dialogue acts is reasonably tractable. More complex approaches have also been developed.[Mairesse and Walker, 2007]. The most common approach used for the text to speech synthesis is the unit selection approach, which splices segments of speech from a database to generate sound for a given word sequence and other method is based on Hidden Markov's model.

The process iterates until one of the conversant (user or machine) terminates the dialogue.

2.8 User Simulation

It is essential to test a Spoken Dialogue System with user dialogues but it is a difficult and time consuming exercise to generate the possible dialogues and then test the dialogue manager with human users. Simulated environments provide one way of speeding up the development process by providing a more efficient testing mechanism.

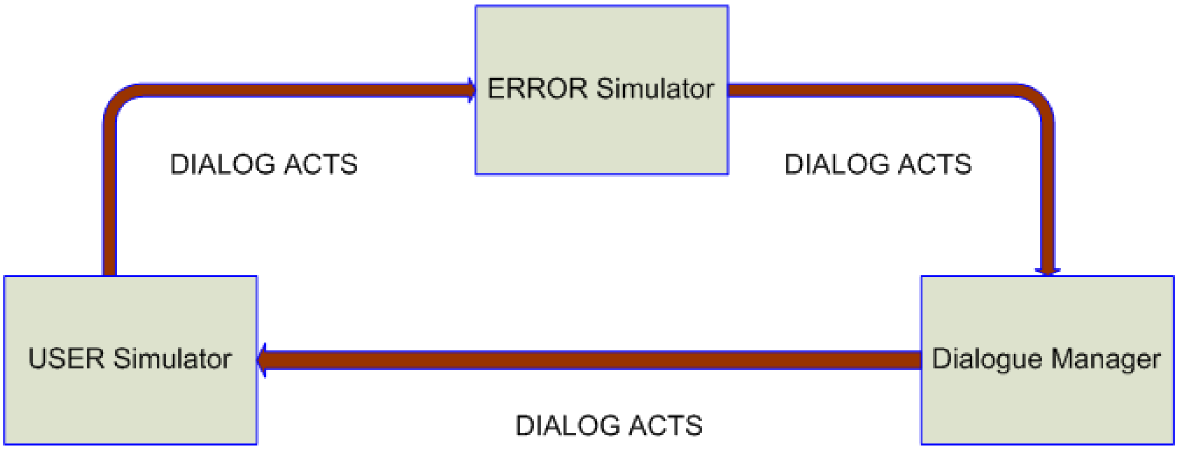


Figure 2.3: Dialogue Act showing User simulator instead of User - A Graphical representation showing the user simulator and error simulator instead of the user in the dialogue act.

A simulated environment generates situations that the dialogue system designer will not have thought about and the system can be refined to handle them. Dialogue managers that are built using techniques from machine learning can learn automatically what actions to take and for these systems simulated environment is particularly important as the system can be boot-strapped by learning from interactions with the simulator. Further refinements obtained from real interactions with human users make the dialogue manager act like humans do in various situations.A user simulator generates dialogue acts given the past dialogue history, as if it were human. This is passed through an error simulator which generates appropriate confusions and confidence scores. The user simulator operates on the dialogue act level as shown in figure 2.3 which graphically represents the user simulator instead of the human user in the dialogue act.

There are also simulation environments which have been built to operate at a word-level [Jung et al., 2009], [Schatzmann et al., 2007]. In this case, the simulated dialogue act is used to generate a word-level form, which is passed to the error-simulator to produce word-level confusions. This is then passed to the semantic decoding component of the spoken dialogue component as in the case of the human machine conversation.

The data-driven simulation techniques which are available and used for user simulation are as follows :

- Bigram models.
- Goal-based models.
- Conditional random fields (CRF Models).
- Hidden agenda models.

A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies is given in [Schatzmann et al., 2006].

2.9 Dialogue Manager Design Paradigms

In Spoken Dialogue System design, the most challenging task is to build an effective dialogue manager which can deal with the uncertainty as well follow policies which can optimally learn from the dialogue history automatically. There are four different paradigms for building the dialogue manager as given below.

2.9.1 Hand Crafted Approach

The dialogue manager at the most basic level can be defined by, the concepts of belief state, state transitions and policy In the hand crafted dialogue management framework the system designer directly define all of these components. As dialogues become more complex, the number of states, transitions, and policy decisions becomes very large so researchers have developed various techniques to facilitate the design process.

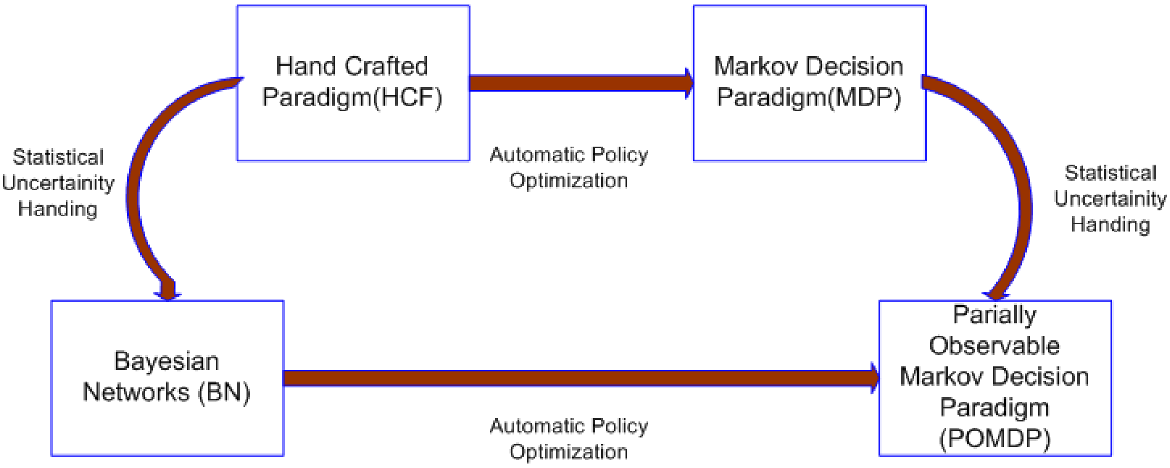

Hand Crafted Paradigm(HCF)
Markov Decision Paradigm(MDP)
Automatic Policy Optimization
Statistical Uncertainity Handing
Statistical Uncertainity Handing
Bayesian Networks (BN)
Parially Observable Markov Decision Paradigm (POMDP)
Automatic Policy Optimization

Figure 2.4: Four major paradigms in spoken dialogue management - The figure shows four major paradigms in spoken dialogue management viz. hand-crafted approaches (HDC), Markov Decision Process models (MDP), Bayesian Network approaches (BN) and Partially Observable MDP approaches (POMDP) with respect to how they handle uncertainty and policy optimisation

The simplest approach to represent the dialogue manager is with the help of a graph or flowchart, sometimes called the callflow [Pieraccini and Huerta, 2005]. Nodes in the graph represent belief states of the dialogue and define which action should be taken, while transitions are determined by the observations received. This approach has proven to be very effective when the system's prompts elicit highly restricted responses from the user. On the other hand, the call-flow model typically struggles when users take the initiative and direct the dialogue themselves. Another approach is frame-based dialogue manager, also known as a form-filling dialogue manager [Goddeau et al., 1996]. Frame-based approaches assume a set of concepts that the user can talk about, called slots, which take on values from a pre-defined set. The current set of filled slots is included as part of the state, along with some hand-crafted information about how certain the system is about each slot value. Dialogue management proceeds by using a pre specified action for each set of known slots. The ability for users to speak about any slot at any time allows for much more freedom in the dialogue, which is typically perceived as more natural for users. The frame based approach is most often used in information seeking dialogues, where a user is seeking information subject to a set of constraints. e.g health information system, which might have slots for the symptoms, time of start of the problem. The system would ask for slot-values until it decided that enough have been given, at which point it would offer information about a relevant doctor. A problem with frame-based approaches is that some of the slots will not be relevant for particular dialogues. Another issue is that dialogues are often composed of smaller sub-dialogues, and the progress in a dialogue often follows a particular agenda, which cannot easily be modelled using frames. This has led to the development of hierarchical and agenda-based dialogue managers. These approaches allow the definition of sub-dialogues which depend on the values of slots at higher levels. When the domain of discourse moves away from information-seeking tasks, then the agenda-based and frame-based approaches sometimes struggle to adapt.

Dialogue systems which are based on shared plans and not only what the user wants such as collaborative dialogues e.g. language learning, where both participants must work together to achieve a task are said to use *plan-based dialogue managers* [Rich and Sidner, 1998]. Another framework for dialogue management is the information state model [Bos et al., 2003] where in the dialogue acts correspond to dialogue moves and are used to update an information state subject to certain preconditions. The information state represents the accumulation of everything that has happened in the dialogue, and is used by the dialogue manager to choose it's next action according to an update strategy such as logic programming [Fodor and Huerta, 2006].

All the above framework's help in structuring the dialogue manager but these approaches don't help in analyzing how to handle the uncertainty which is inherent in the dialogue process.

2.9.2 Sequential Decision Process approach

Spoken Dialogue Systems have to deal with uncertainty which is inherent during the process of speech recognition and natural language interpretation. A Common approach is to augment the hand crafted belief states with states which represent uncertainty as well [Bohus and Rudnicky, 2005]. But this model lacks the principled definition for these states of uncertainty, so the alternative was sequential decision process models which were natural and augmented a well researched framework based on Markov Decision Processes(MDP's) [Puterman, 1994], to aid the spoken dialogue system to reliably identify the underlying environment state.

A spoken Dialogue System based on the *Sequential Decision Process framework* interacts synchronously with the external environment i.e. the user with the main goal of maximizing its *reward* by taking appropriate *actions*. These actions and history of the environment states determine the probability distribution over next possible states and as such are modelled as a stochastic process.

2.9.2.1 Markov Decision Processes framework(MDPF)

A formal model of fully-observable sequential decision processes which is an extension of Markov chains with a set of decisions/actions and a state based reward structure.In this process for each state a decision has to be made regarding the action to be taken in that state to increase some predefined measure of performance. The action affects not only the transition probabilities but the rewards as well. A state describes the environment at a particular instant of time.In this thesis it is assumed that the system can be in a finite number of states and the agent(Spoken Dialogue System) can choose from a finite set of actions. Let $S = s_0, s_1, s_2, ..., s_N$ be a finite set of states. Each state at discrete time $t \in T$ is viewed as a random variable S^t whose domain is the state space S as the process is stochastic. The past history in the form of system states is irrelevant in predicting the future so the state must contain enough information to predict the next state for the process to be *Markovian*

$$Pr(S^{t+1}|S^0, S^1, ...S^t) = Pr(S^{t+1}|S^t) \tag{2.1}$$

The Spoken Dialogue System at each state execute one of the available action (a) from a set of actions (A) which affects the state transition probabilities. Thus each action $a \in A$ is fully transcribed by a $|S|X|S|$ state transition matrix whose entry in i^{th} row and j^{th} column is the probability that the system will move from state s_i to the state s_j if the action a gets executed.

$$P^a_{ij} = Pr(S^{t+1} = s_j|S^t = s_i, A^t = a) \tag{2.2}$$

The effect of the actions A on the system states S is given by *transition function* (T) where $T : SXA \rightarrow \Delta(S)$ which associates a probability distribution over the possible successor states. and $\delta(S)$ represents the set of probability distribution over S. Thus for each state s, s' and $a \in A$ the function T determines the probability of a transition from state s to state s' after executing action a.

$$T(s, a, s') = Pr(S^{t+1} = s' | S^t = s_i, A^t = a) \tag{2.3}$$

The spoken dialogue system assigns a reward (or cost if the value is negative) for being in a state s and executing action a using a reward function $R : SXA \rightarrow \mathbb{R}'$). The casual relationship between MDP states, actions and rewards is shown in the figure 2.5

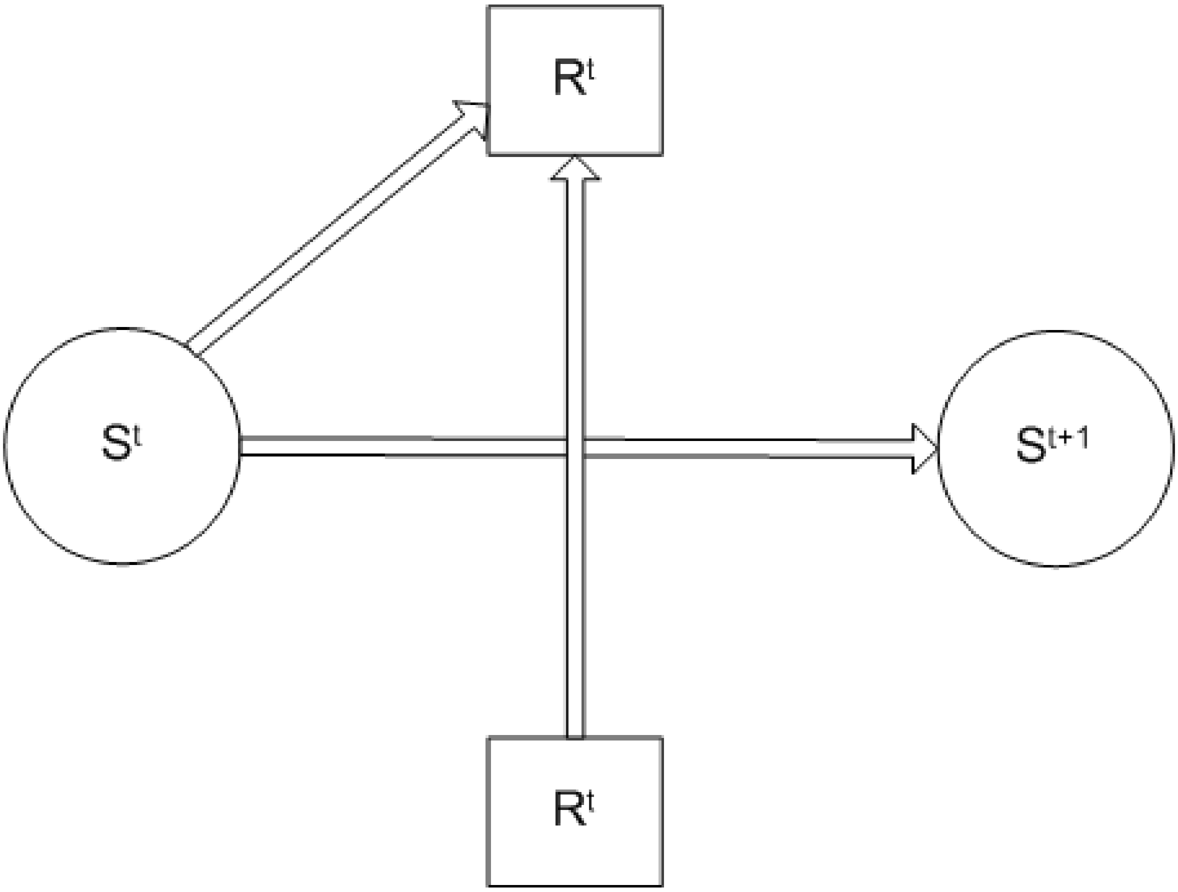


Figure 2.5: Casual Relationships between MDP states, actions, rewards - R^t is the reward received at time t

The Markov Decision Process was first suggested as a dialogue model by [Levin and Pieraccini, 1997]. The system proposed used bi-gram language model for training and optimized the reward using standard algorithms. [Walker, 2011] proposed an MDP based system (PARADISE framework) by using regression on the known features of the dialogue as a means to determine the reward the system should assign during each turn of the dialogue. This system along with various other dialogue systems have been successfully tested with the human users [Kearns et al., 2011]. But since the state transitions are mostly handcrafted by the system design spoken dialogue system based on MDP framework, the state set gets intractable when the complexity of the dialogue increases. For example the information state updates have been used in the MDP systems by adding the concept of rewards and Markov property(i.e. the system belief state depends only on its previous value and not on the history) to the standard information state model [Lemon et al., 2006]. There is some research work [Paek and Chickering, 2006] indicating that this may not necessarily be a valid assumption.

2.9.3 Partially Observable Markov Decision Process (POMDP) Framework

In order to act optimally, the spoken dialogue system must take all the previous history of observations and actions into account, rather than just the current sate it is in. A POMDP is a generalization of MDPs in which system states are not fully observable.Partially observable Markov Decision Process(POMDP) were first suggested for dialogue by [Roy et al., 2000]. A POMDP framework is based on the underlying MDP extended with observation space O and observation function $Z(.)$. In MDPs the dialogue system has the complete knowledge of the system states whereas in case of partially observable environments, observations are only probabilistically dependent on the underlying environment state. Also the same observations can be observed in different states which makes it difficult to determine the state of the system. *Observation function* $Z : S \times A \rightarrow \delta(O)$ specifies the relationship between the system states, actions and the observation space. Thus $Z(s', a, o')$ is the probability that observation o' will be recorded after an agent performs action a and moves to state s'. Thus

$$Z(s', a, o') = Pr(O^{t+1} = o' | S^{t+1} = s', A' = a) \tag{2.4}$$

Formally POMDP is a tuple $< S, A, T, R, O, Z >$ where S is the set of states, A is the action space, $T(.)$ is the transition function, $R(.)$ the reward function, O is the observation space, and $Z(.)$ is the observation function. The casual relation between the elements of the tuple are shown in the figure 2.6

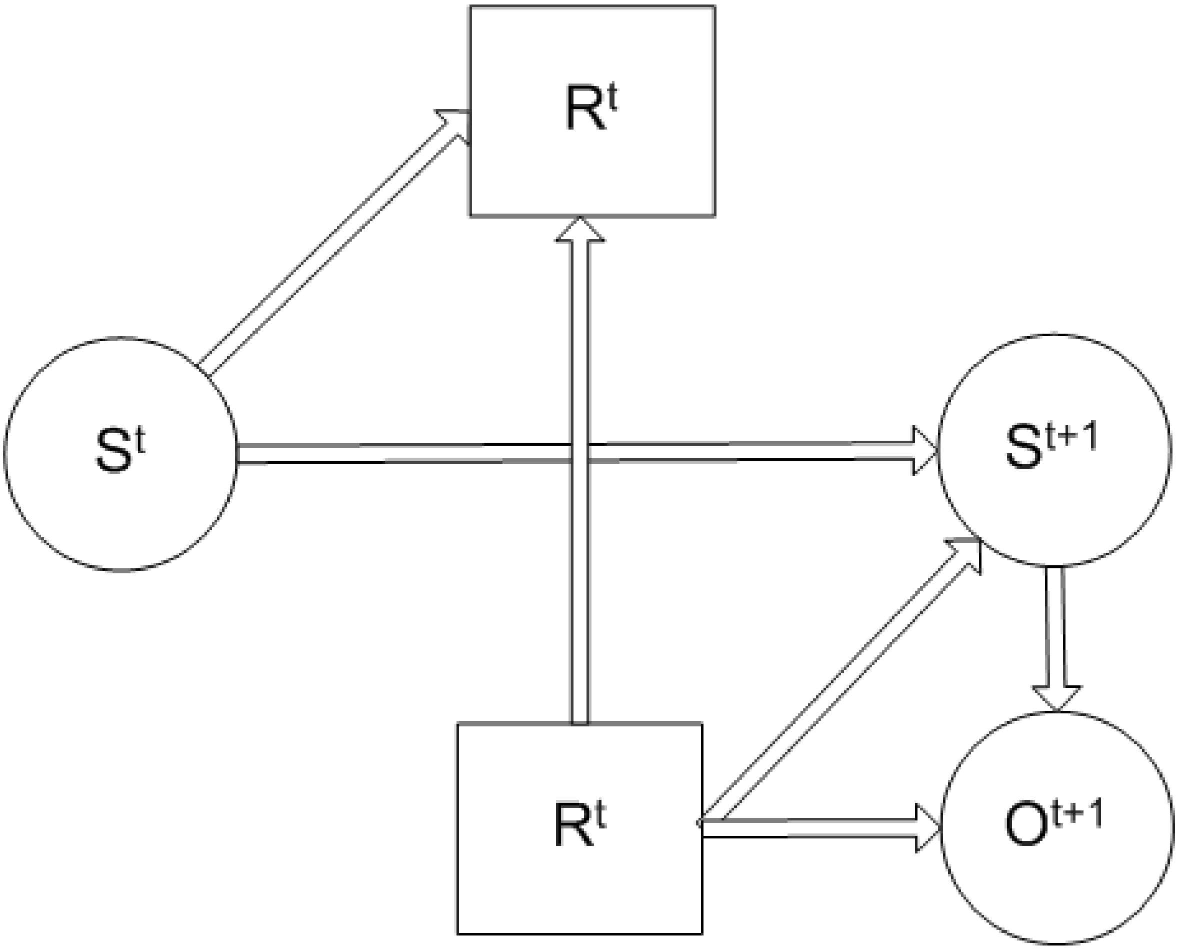


Figure 2.6: Influence Diagram in a POMDP framework - Casual Relationships between POMDP states, actions, rewards and observations

2.9.3.1 Process History

In a POMDP the complete system history from start till time t is represented by a triplet i.e. by the system state, the observation and the action taken e.g. $(s^0, O^0, A^0), (s^1, O^1, A^1), .$ The history is the record of everything that has happened during the execution of the process. In partially observable environment, the system bases its decision on the observable history as it cannot fully observe the underlying world state. The SDS has the prior belief about the world state which are summarized by the probability distribution

b_0 over the system states and the system starts by executing an action a_0 based on the distribution b_0. The set of all observable histories or trajectories are represented as H_0. Representing and structuring hH_0 in different ways has led to different POMDP solutions and *Policy* execution algorithms.

2.9.3.2 Performance Measures

The system trajectories are ranked with the help of a *Value function* ($V : H \to \mathbb{R}$) which assigns a real number to each system history $h \in H$. A history h will be preferred over h' if $V(h) > V(h')$. In case of infinite horizon problems i.e. the problems where the decision stops after a finite number of steps the value function for a system trajectory h of length l is simply the sum of rewards attained at each stage [Bellman, 1954]

$$V(h) = \sum_{t=0}^{t=l} R(s^t, a^t) \tag{2.5}$$

In case of infinite horizon problems i.e. the problems where the system trajectory is unbounded, a discount factor γ is introduced which states that the rewards received later get discounted which contribute less than current rewards.The value function for such total discounted reward function is given as [Bellman, 1954]

$$V(h) = \sum_{t=0}^{\infty} \gamma^t R(s^t, a^t) \tag{2.6}$$

2.9.3.3 Policy

On each turn in a spoken dialogue, the system has to decide and execute an optimal course of action in an uncertain environment contingent on the observable history. A *policy* $\pi : H_0 \mapsto A$ is a rule that maps observable histories into actions. The main aim of the spoken dialogue system is to choose a policy which maximizes the objective function that is defined on the set of system trajectories(H_0). Given a history

$$h' =< a^0, o^0 >, < a^1, o^1 >, ..., < a^{t-1}, o^t >$$

the action prescribed by the policy π at time t would be $a^t = \pi(h^t)$ where a^0 is the system's initial action and o^t is the latest observation.

The likelihood of particular system trajectory is controlled by inducing the probability distribution $Pr(h|\pi, b_0)$ over all possible sequence of states and actions by the system for initial distribution(b_0). The *Expected policy value* is the expected value of system trajectories induced by the policy π and is given by

$$EV(\pi) = V^{\pi} = \sum_{h \in H} V(h) Pr(h|\pi, b_0) \tag{2.7}$$

The system's goal is to find a policy $\pi^* \in \Pi$ with the maximum expected value from the set Π of all possible policies. The policies are generally represented using tractable representations where in the observable histories are either represented as probability distributions over system states or grouped into a finite set of distinguishable classes using finite-suffix trees or Finite state controllers.

2.10 Conclusion

This chapter discusses the importance of spoken language in the design of the human computer interaction process. When a user interacts with a computer, the user unknowingly embeds lots of information at different levels of speech which if capitalized properly can help in the design of an efficient and effective dialogue system. The chapter also elaborated on different modules that are to be focussed on during the design of a spoken dialogue system. Testing of a spoken dialogue system with different dialogues is a challenging task, the system simulated user dialogue for testing purpose has shown remarkable results in the design of a spoken dialogue system. Dialogue manager forms the heart of the Spoken Dialogue system and the strategy it follows to reply to user query/utterance has to be based on some prior knowledge of the dialogue context. Various approaches in the design of the dialogue manager has been discussed in the last section.

CHAPTER 3

DIALOGUE, REPRESENTATION, INFERENCE AND LEARNING STRATEGIES

In the last many years researchers have designed spoken dialogue system which have the capability to communicate with the users in the real time. Many Spoken Dialogue systems are realized which are the good examples of real time, goal-oriented interactions between humans and computers that perform tasks like finding a good restaurant nearby, reading your email, perusing the classified advertisements about cars for sale, or making travel arrangements (Seneff, Zue, Polifroni, Pao, Hether- ington, Goddeau, & Glass, 1995; Baggia, Castagneri, & Danieli, 1998; Sanderman, Sturm,den Os, Boves, & Cremers, 1998; Walker, Fromer, & Narayanan, 1998). Yet in spite of 40 years of research on algorithms for dialogue management in task-oriented dialogue systems, (Carbonell, 1971; Winograd, 1972; Simmons & Slocum, 1975; Bruce, 1975; Power, 1974; Walker, 1978; Allen, 1979; Cohen, 1978; Pollack, Hirschberg, & Webber, 1982; Grosz, 1983; Woods, 1984; Finin, Joshi, & Webber, 1986; Carberry, 1989; Moore & Paris, 1989; Smith & Hipp, 1994; Kamm, 1995) inter alia, the design of the dialogue manager in real-time, implemented systems is still more of an art than a science (Sparck-Jones & Galliers, 1996). This chapter discusses a method, and experiments that validate the method, by which a spoken dialogue system can learn from its experience with human users to optimize its choice of dialogue strategy.

The dialogue manager of a spoken dialogue system accepts the user's utterance which is represented as a frame of Spoken Language Understanding modules results and then chooses in real time what information to communicate to the human user at a conceptual level and how to communicate it. The choice it makes is called its 'strategy' The system responses have to reflect the discourse context by maintaining the discourse history.

The dialogue manager can be formulated as a state machine, where the state of the dialogue is defined by a set of state variables representing observations of the user's conversational behaviour, the results of accessing various information databases, and aspects of the dialogue history. Transitions between states are driven by the system's dialogue strategy. However, There are a many possible choices for policies at each state of a dialogue. Decision theoretic planning can be applied to the problem of choosing among dialogue strategies, by associating a utility U with each strategy (action) choice and by positing that spoken dialogue systems should adhere to the **'Maximum Expected Utility Principle'** which states that an optimal action is one that max-imizes the expected utility of outcome states (Keeney & Raiffa, 1976; Russell & Norvig, 1995), Thus, a SDS can act optimally by choosing a strategy (a) in state

S_i that maximizes $U(S_i)$. Several reinforcement learning algorithms based on dynamic programming specify a way to calculate (S_i) in terms of the utility of a successor state S_j (Bellman, 1957; Watkins, 1989; Sutton, 1991; Barto, Bradtke, & Singh, 1995), so if the utility for the final state of the dialogue were known, it would be possible to calculate the utilities for all the earlier states, and thus determine a policy which selects only optimal dialogue strategies. Previous work suggested that it should be possible to treat dialogue strategy selection as a stochastic optimization problem in this way (Walker, 1993; Biermann & Long, 1996; Levin, Pieraccini, & Eckert, 1997; Mellish, Knott, Oberlander, & O'Donnell, 1998). There are three main possibilities for a simple reward function: user satisfaction, task completion,or some measure of user effort such as elapsed time for the dialogue or the number of user turns. But it appeared that any of these simple reward functions on their own fail to capture essential aspects of the system's performance. For example, the level of user effort to complete a dialogue task is system, domain and task dependent. Moreover, high levels of effort, e.g., the requirement that users confirm the system's understanding of each utterance, do not necessarily lead to concomitant increases in task completion, but do lead to significant decreases in user satisfaction (Shriberg, Wade, & Price, 1992; Danieli & Gerbino, 1995; Kamm, 1995; Baggia et al., 1998). Furthermore, user satisfaction alone fails to reflect the fact that the system will not be successful unless it helps the user complete a task. A method for deriving an appropriate performance function was a necessary precursor to applying stochastic optimization algorithms to spoken dialogue systems in the paradise method for learning a performance function. In this chapter, we apply the paradise model (Walker, Litman, Kamm, & Abella, 1997a) to learn a performance function from a corpus of human-computer dialogues, which we then use for calculating the utility of the final state of a dialogue in experiments applying reinforcement learning to selection of dialogue strategies.

3.1 Dialogue State Representation

In a spoken dialogue system, the dialogue manager form the heart of the system. After the utterance is converted into a natural language form and the meaning is interpreted by natural language module, the dialogue manager has to decide how and what to say to the user to fulfil the system's turn in the dialogue. In order to obtain computationally efficient algorithms, the structure of the domain under consideration must be exploited. The Dialogue Manger in the HBIS has to maintain the *state* which is defined by a set of state variables that represent the information that has happened during the dialogue process and aids the dialogue manager in deciding the what the system should do in opposite to the user utterances.The state variables encode various observations of the user conversational behaviour, such as results of processing speech with the natural language understanding module and results from accessing information databases relevant to the application as well as certain aspects of the current context. The better way to represent the state is using a probabilistic approach as a belief distribution over environment states and updated using Bayes theorem.

3.1.1 Probabilistic Graphical Models

Probabilistic Graphical Models are graphs in which nodes represent the random variables and the arcs represent the conditional independence assumptions and thus provide a representation for the joint probability distributions. A graphical model needs fewer parameters based on the conditional assumptions and thus fit for efficient inferencing and learning when compared to other representations.

There are three types of graphical models:-

1. **Directed Graphical Models** also known as Bayesian Networks, Belief Networks, Generative models, Casual models etc are graphs in which the arc are directed and are mostly used in machine learning applications.

2. **Un-Directed Graphical Models** also known as Markov networks, Markov Random Fields are graphs in which the arc are undirected.

3. **Chain Graphs** are models in which the arc are both directed and undirected.

3.1.2 Bayesian Networks

Its a directed acyclic graphical model which give an intuitive representation for the various assumptions and belief states in a system and also facilitate the use of computationally effective algorithms for updating the beliefs in an environment state whenever an observation is made. A Bayesian Network is a representation for statistical models where each node represents a random variable and the edges represent the probabilistic constraints between edges. If an arc between two nodes X and Y is interpreted as "X causes Y". The joint distribution of all variables in the graph factorises as the product of the conditional probability of each variable given its parents in the graph. In a POMDP based framework the assumptions are represented by the network as shown in the figure 3.1

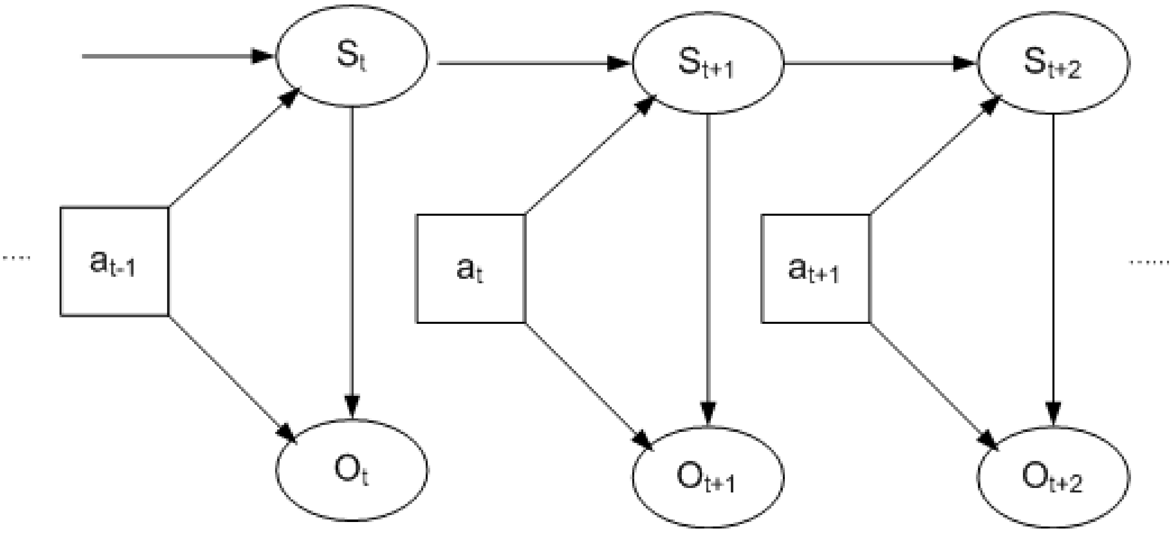


Figure 3.1: A portion of Bayesian network representing the POMDP Model
-

Networks as shown in the figure which repeat the structure(*time-slices*) at each interval in time are refereed as *Dynamic Bayesian Networks.* Actions of the system(a_t) are shown in the rectangles and Shading of Observation nodes(O_t) represent that they are observed. The Bayesian networks for dialogue allow further factorization of the dialogue system environment state (S_t) which aids the updation of factor beliefs using various efficient algorithms.[Williams and Young, 2005] factorized the environment state into three components $s_t = (g_t, u_t, h_t)$ where g_t represents the long term goal of the user, u_t represents the true user act and h_t represents the dialogue history.Further structuring can be done by representing the state into $slotsc \in C$ where slot implies a concept for which the user must specify a value e.g In a TIS the concept may be destination or type of accommodation or food. The state is thus factorised into sub-goals $g_t^{(c)}$ or sub-histories $h_t^{(c)}$. The sub-histories $h_t^{(c)}$ depend on the user act U_t and the previous sub-histories $h_{t-1}^{(c)}$. The user act depends on the set of sub-goals $g_t^{(c)}$.

3.1.2.1 Dependencies

The exists a strong dependency between the concepts of the real world dialogue as the user intention clarifies. Given the user action, the sub-goal nodes cannot be assumed to be independent as such there will be a dependency which is to be limited to enable tractability whereas the sub-history nodes can be independent.A method to limit the dependencies is to add the *validity node* $[v_t^{(c)}]$ for each concept which indicates whether the associated sub-goal is relevant to the overall user goal or not. The validity node takes the value either 'Applicable' or 'Not Applicable'. If the validity node is 'Applicable' then the sub-goal also become applicable and the user sub-goal depends on the previous value with some probability of change.

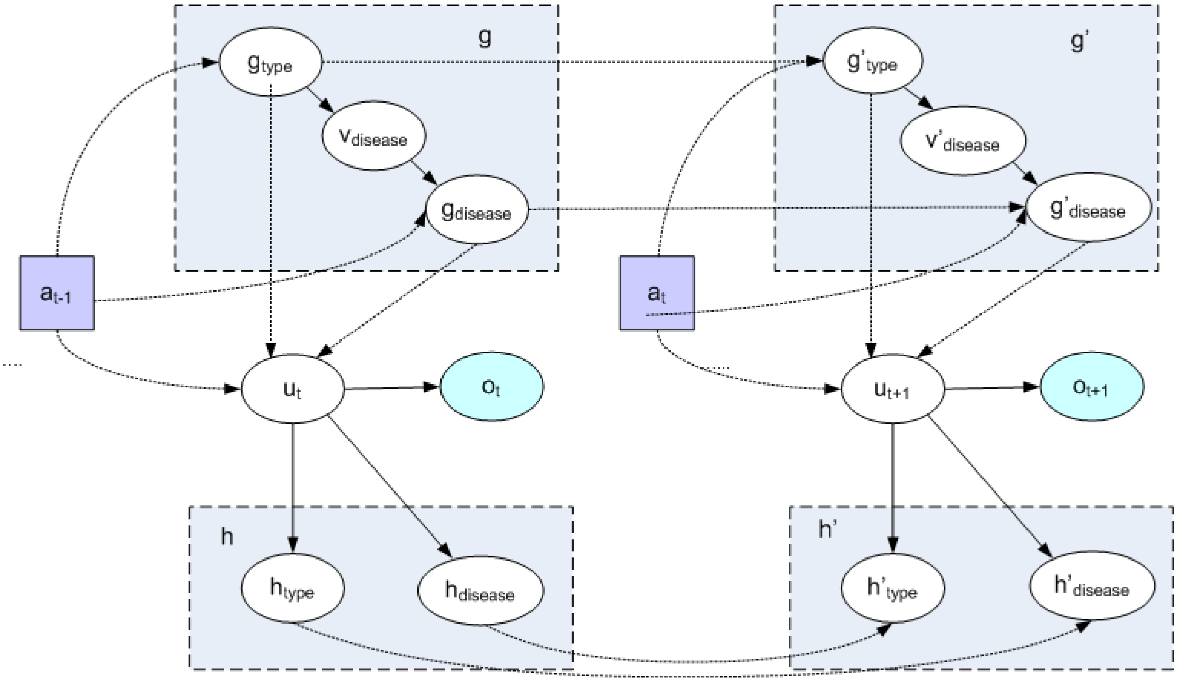


Figure 3.2: Factorisation of Bayesian network representing part of a health based information system - $v_disease$ is the validity node for disease, g_type represents the type of department being sought

Figure 3.2 shows the network for a health based information system representing two types of concept viz. the type of department the patient wants to visit and the disease the patient is suffering from. The user act and user goal are assumed to be independent from the previous history. When a patient asks for the type of department, the disease concept may not be applicable. But once the patients talks about the disease it becomes relevant and hence applicable.Thus it clearly indicate the intention of the user that he wants to visit a particular department for a disease he mentions and hence the validity of the disease node increases.

3.1.2.2 History Nodes

History nodes store the information about the acts that have happened in a dialogue and is used for framing a dialogue strategy. The sub-history is separated into the what the user wants/desires to know $d_t^{(c)}$ and the grounding information for each concept $i_t^{(c)}$.The desire variable $d_t^{(c)}$ may requires only few values such as NOTHING SAID, REQUESTED, INFORMED. This allows the system to record when the user requests for the value of a concept. The grounding information nodes $i_t^{(c)}$ stores the last grounding state for the concept value.

3.1.2.3 User Acts and Observation Nodes

In a real world spoken dialogue system there are large number of state variables and updating the belief states of these variables in a Bayesian network will be computationally expensive.In such situations, the user acts are split for each concept represented as $u^{(c)}$ and depends on the user goal for that concept. Similarly the observation nodes are split into sub-observation $o^{(c)}$ which store how a observation is related to a given concept. Actions that do not apply to any concept appear in all concept-level observations.

	Dialogue Act	Confidence Score
Overall Observation	inform(type=orthopaedics, disease=fracture)	0.9
o;	inform(type=orthopaedics, disease=pain)	0.1
Type Observation o_{type}	inform(type=orthopaedics)	1.0
Disease Observation	disease=fracture	0.9
$o_{disease}$**:**	disease=pain	0.1

Table 3.1: Split of overall observation into concept level observations

Since the observations are completely factorised, the approach may not give better performance. However this approach can be used for modelling highly complex user action models with the help of independent concept level user acts for which different user action probabilities are used for each concept which are then joined to determine the probability of the overall user action.

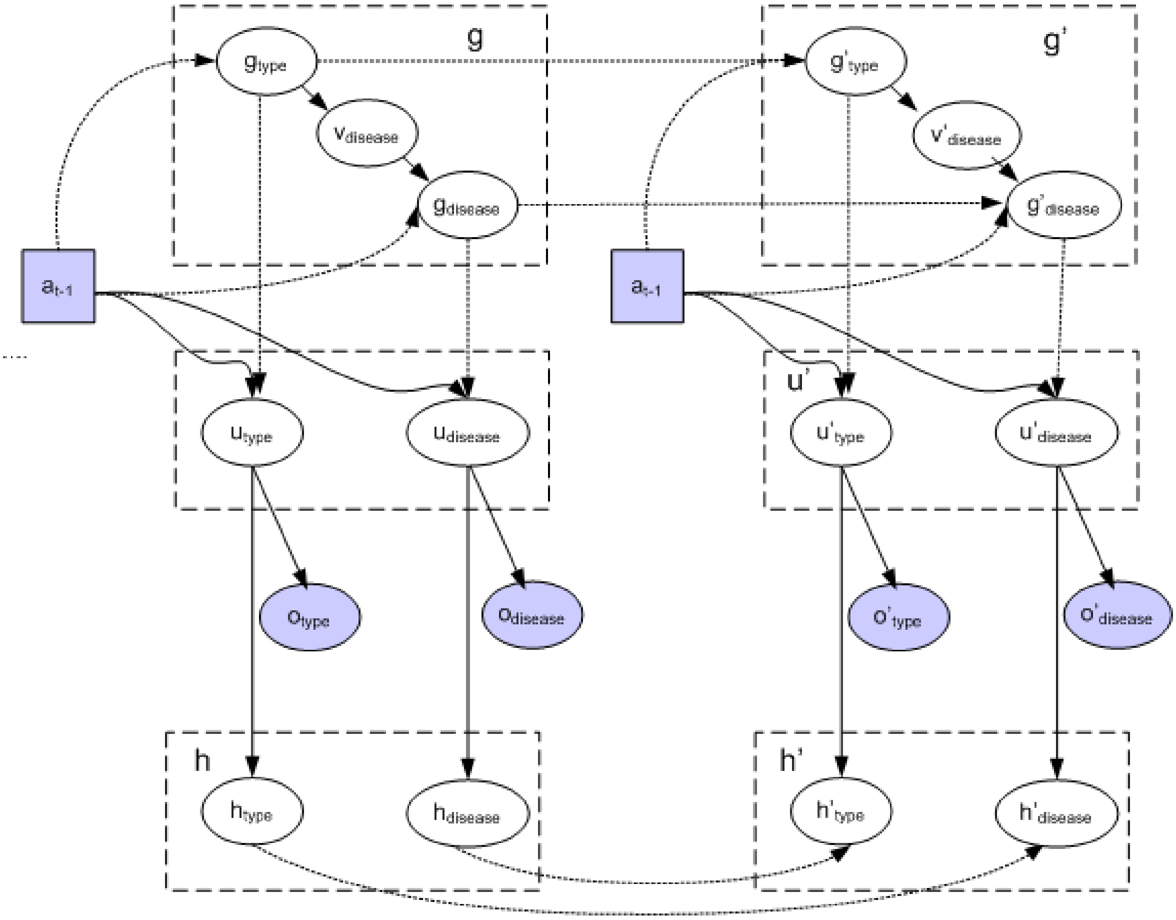


Figure 3.3: Bayesian network with splitted user-acts -

3.2 Factor Graphs

Factor Graphs (fgraphs) is a representation which unifies directed and undirected graphs [Kschischang et al., 2001] proposed a graphical framework known as *Factor graphs* which provides a suitable formalism for analysing the independence of variables in a spoken dialogue system. Also there exists many efficient algorithms for updating the beliefs which can be used by the dialogue manager. Factor graphs are undirected bipartite graphs comprising of two types of nodes which represent random variables(Circle nodes) and factors (Square node). Factor Graphs are bipartite because each variable node X_i is connected to all the factor nodes F_i which contain X_i in their domains.

In general factor graphs specifies how a function of many variables can be decomposed into a set of local functions. *Factors* are not probabilities themselves, they are functions that determine all probabilities. The joint distribution over all random variables can be written as a product of factor functions, one for each factor node. These factors are a function of only the random variables connected to the factor node in the graph.There is a direct mapping from Bayesian networks to factor graphs. Figure 3.4 is a factor graph representation of the POMDP assumptions, previously depicted as a Bayesian network in Figure 3.1

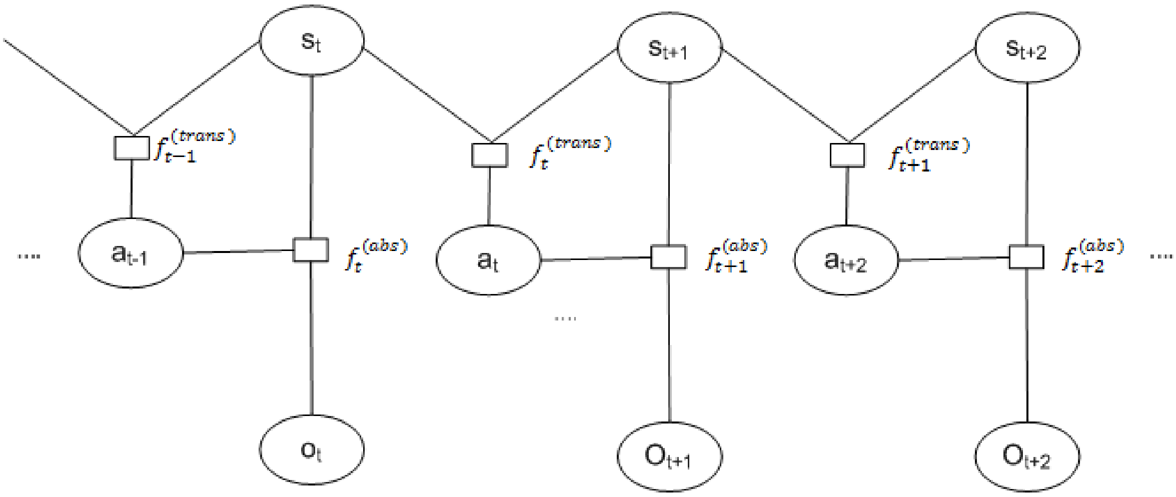


Figure 3.4: Factor graph representing POMDP model -

In this factor graph, the environment state transition function is represented as $f_t^{(trans)}$ and thus $f_t^{(trans)}(s_t, s_{t+1}, a_t) = P(s_{t+1}|s_t, a_t)$ Also $f_t^{(obs)}$ represents the observation function $f_t^{(obs)}(s_t, o_t, a_{t+1}) = P(o_t|s_t, a_{t-1})$ The variables in the factor graph are denoted by X_i and the variable values by x_i,the factors by f_β and vector $x = (x_1, x_2, ...x_{N_i})$ represent the variable values simultaneously. Each factor will depend on a a subset of random variables. As such they can be defined as functions over the whole set denoted by $f_\beta(x)$ which is achieved by defining factor values as factor values of the original subset. Thus the joint distribution in the factor graph factorises as

$$p(x) \propto \prod_\beta f_\beta(x) \tag{3.1}$$

3.3 Loopy Belief Propagation Algorithm

3.3.1 Belief Propagation

Belief propagation is a way of computing exact marginal posterior probabilities in graphs with no undirected cycles (loops). The system takes an action based on the current set of beliefs in the environment state which must be updated whenever an observation is made.The decision can be based on the marginal distribution of the beliefs over a single variable and is computed by integrating or summing out all other random variables from the joint distribution. The marginal distribution provides the information that can aid the dialogue management decision and thus save the computational time. The $\tilde{x}_1$ notation indicates that the variable X_1 is fixed while the other variables are either integrated or summed thus the marginal $p(\tilde{x}_1)$ would be computed as

$$\begin{aligned} p(\tilde{x}_1) &= \int p(\tilde{x}_1, x_2, ..., x_{N_i}) dx_2, ..., dx_{N_i} \\ &= \sum_{x_2,....,x_{N_i}} p(\tilde{x}_1, x_2, ..., x_{N_i}) \\ &= \sum_{\mathbf{x}:x_1=\tilde{x}_1} p(\mathbf{x}) \end{aligned}$$

Belief Propagation provides exact inferencing when there are no loops in graph (e.g. chain, tree.) It is equivalent to dynamic programming/Viterbi in these cases. The marginal distribution is generally not tractable as the belief propagation becomes exponential in the size of the nodes. so an approximate algorithm like the Loopy Belief Propagation(LBP) or sum-product algorithm [Kschischang et al., 2001] is to be used to enable tractability.

When loops are present in the network the network is no longer single connected. The local propagation schemes may not work and will run into trouble. If the loops are ignored, and permit the nodes to communicate with the factors, the message will circulate around the loops and the process may not converge to a stable equilibrium. Loopy propagation algorithm maintains a set of messages for each arc in the model. For each arc between a node representing a random variable X_i and a factor f_a there are two defined messages. $\mu_{X_i \to f_a}(x_i)$ is a message from the variable to the factor and $\mu_{f_a \to X_i}(x_i)$is the message from the factor to the random variable. Both of these are the functions of the possible values of X_i. Once the messages are computed the marginal probability of a random variable X_i is calculated from the message to that variable from the neighbouring factors, $a \in ne(X_i)$ If k is the normalizing constant then

$$p(x_i) = k \prod_{a \in ne(X_i)} \mu_{f_a \to X_i}(x_i) \tag{3.2}$$

Algorithm 3.1 Loopy Belief Propagation Algorithm

```
**Intialize :** Set all messages equal to one.
Let $Y = \{\mathbf{x} = (x_1, x_2, ..., x_N)^T | x_i \text{ is the possible value of } X_i\}$
repeat
  Choose a factor $f_a$ to update. Suppose this is connected to variables $X_1, X_2, ..., X_N$.

  First update the approximation as follows :
  for each variable $X_i$ connected to the factor  do
    Update the message out of the factor
    $(\forall x_i^{'})\mu_{f_a \to X_i}(x_i^{'}) = \sum_{x \in Y, x_i = x_i^{'}} \prod_{j \neq i} \mu_{X_j \to f_a}(x_j)$.
    Update the cavity distributions
    Update the message into nearby factors
    for  each factor $b \neq a$ connected to variable $X_i$  do
      $(\forall x_i^{'})\mu_{X_i \to f_b}(x_i^{'}) = \prod_{a \neq b} \mu_{f_a \to X_i}(x_i^{'})$.
    end for
  end for
until convergence
```

The iterative process of belief updates using factor graphs continues until the approximate distribution no longer changes significantly, at which stage the algorithm is said to be converged and the resulting set of messages will constitute a fixed point of the algorithm. If the factor graph has a tree structure, the algorithm will converge after a breadth first traversal and followed by a reverse sequence of updates.

3.4 Limiations

In a dynamic bayesian network like POMDP the number of nodes grows with time. In order to update its beliefs the system will have to maintain all the information for all the nodes in the network i.e. for the most recent time-slice, the approximations for all the previous time-slices are needed. This issue contradicts the MArkov property of POMDP i.e. the beliefs of the current time slice depends on the beliefs at the previous time-slice. The Loopy Belief Propagation algorithm computes approximate marginal distributions and if the network is highly connected, the observations from future time-slices may effect the computation of marginals of the previous time-slices which then affects the computation of marginals at the current time-slice. Thus it will be a added responsibility for the system to store the information related to the cavity and factor approximation for all nodes for the entire duration of the dialogue. To overcome the difficulty of recomputing the approximation for all previous time-slices for the lengthy dialogue, it is preferable to limit the number of time-slices that are maintained [Murphy, 2002] Given a number n, the factor updates are limited to n time-slices, the marginals for variables connected to factor nodes are computed at time t by approximating the joint distribution at time $t-1$ for the most recent n time-slices and are maintained for future updates. [Boyen and Koller, 1998] suggested an approach wherein only the marginal approximation at time $t-1$ is used to compute the exact marginal distributions of the current time slice which are then stored and used for the next iteration.

3.5 Expectation propagation

Complex Spoken Dialogue System have to deal with a large state spaces and when the nodes have a large number of values it becomes difficult to update the beliefs using Loopy Belief Propagation Algorithm.In case of arbitrary approximation, the marginal matching can be replaced by minimizing a distance function between two probability distribution known as divergence measure. Expectation propagation [Minka, 2001] is like belief propagation except it requires that the posteriors (beliefs) on each variable have a restricted form. Specifically, the posterior must be in the exponential family of the form $q(x) \propto exp(\gamma' f(x))$ where x is a variable. This ensures that beliefs can be represented using a fixed number of sufficient statistics. We choose the parameters of the beliefs such that

$$\gamma^* = \arg D(p(x) \parallel q_\gamma(x))$$

where $p(x)$ and $q(x)$ are two functions for which the distance measure is defined.

$$p(x) = \frac{q^{prior}(x) \times t(x)}{Z}$$

is the exact posterior and $Z = \int_x q^{prior}(x) \times t(x)$ is the exact normalizing constant, $t(x)$ is the likelihood term (message come in from a factor and $q_\gamma(x)$ is the approximate posterior. If the Sequential Bayesian Updation is combined with this approximation after every update step, it is known as Assumed Density Filtering(ADF). or the probabilistic editor which depends on the order in which the update are made on the beliefs.The Expectation Propagation Algorithm being a batch algorithm reduces the sensitivity to ordering by iterations wherein it goes back and re-optimizes each belief in the revised context of the updated beliefs. To achieve this all the messages are to be stored for undoing any of their effect. Thus instead of approximate matching of messages, the posterior are matched using moment matching.Consider the factor graph given in fig 3.6 to understand the difference between the Belief Propagation and Expectation Propagation. A message is sent from f to x and then x belief's are updated as

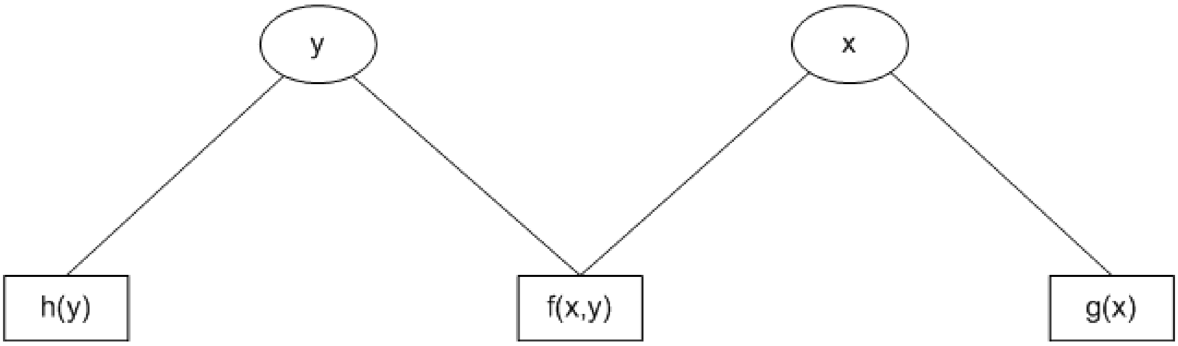


Figure 3.5: A Simple factor graph - Round nodes represent random variables and rectangle nodes represent factors

$$\phi_x^{prior} = \phi_x / \mu_{f \to x}^{old} = \mu_{g \to x}^{old}$$

$$\phi_f^{prior} = \phi_f / \mu_{x \to f}^{old} = f(x.y)\mu_{y \to f}^{old}(y)$$

$$\mu_{f \to x} = \phi_f^{prior} \phi_f \downarrow x = \int_y f(x,y)\mu_{y \to f}^{old}(y)$$

$$\phi_x = \phi_x^{prior} \times \mu_{f \to x} = \mu_{g \to x}^{old} \times \mu_{f \to x}$$

In Expectation Propagation, the approximate posterior ϕ_x is computed first and then the message $\mu_{f \to x}$ is derived which if combined with the prior ϕ_x^{prior} would result in the same approximate posterior.

$$\phi_x^{prior} = \phi_x^{old} / \mu_{f \to x}^{old}$$

$$\phi_f^{prior} = \phi_f / \mu_{x \to f}^{old}$$

$$(\phi_x, Z) = ADF(\phi_x^{prior} \times \phi_f^{prior} \downarrow x)$$

$$\mu_{f \to x} = (Z\phi_x)(\phi_x^{prior})$$

where $(q, Z) = ADF(p)$ produces the best approximation from q to p within a specified family of distributions.

Algorithm 3.2 Expectation Propagation Algorithm

```
For each factor $f$ in order.
$\phi_f = \phi_f^{old}$
for each variable $x$ connected to the factor $f$ in predecessor order do
    Update the message out of the factor
    $\mu_{x \to f} = (\phi_x^{old})(\mu_{f \to x}^{old})$
    $\phi_f = \phi_f \times (\mu_{x \to f})/(\mu_{x \to f}^{old})$
end for
for each variable $x$ connected to the factor $f$ in successor order do
    Update the message in to the factor
    $\phi_x^{prior} = \phi_x^{old} / \mu_{f \to x}^{old}$
    $\phi_f^{prior} = \phi_f / \mu_{x \to f}^{old}$
    $(\phi_x, Z) = ADF(\phi_x^{prior} \times \phi_f^{prior} \downarrow x)$
    $\mu_{f \to x} = (Z\phi_x)(\phi_x^{prior})$
end for
```

3.6 Comparison to previous work

The key feature of the Loopy Belief Propagation is that dialogue manager can deal with complex dependencies between variables by using an approximate updates instead of exact updates.Past Research work of [Bui et al., 2009] which is close to LBP assumes a completely independent factorization of goals and can therefore use the standard Loopy belief Propagation to obtain an exact update. [Young et al., 2010] suggested the Hidden Information State approach which updates probabilities by partitioning the unfactorized state space into group of states which are indistinguishable given the observations. It can be shown that the probabilities for all states in a given partition may be updated simultaneously if the user goal never changes. The use of Loopy Belief

Propagation allows to reduces the computation times by exploiting conditional independence and allow probabilities that the user goal may change.[Henderson and Lemon, 2008] provided an alternative mechanism for state update similar to HIS update based on the Markov Decision Process state mixture. Another approach for belief updates is suggested by [Williams, 2007] and based on particle filters which uses sampling. The approach has shown considerable benefits in terms of computation time of updates when compared to exact updates.

3.7 Grouped Loopy Belief Propagation

Belief updates when the concept takes multiple values improves the computational complexity of the dialogue system. Though Loopy Belief Propagation algorithm exploits the dependencies between the concepts but updating beliefs becomes complex when the concept takes a large number of values e.g the disease concept can takes multiple values in the environment state. The solution to this problem was proposed by [Young et al., 2010] and it suggested to join the indistinguishable environment states into groups.

3.8 Policy Design and Learning

After the system's belief state is defined, the dialogue policy or strategy π which means how the actions are taken is to be defined. In case of Spoken Dialogue system we propose to use reinforcement learning to optimize the policy in which the reward function is defined by the function $r(b, a)$ which indicates the reward obtained by taking the action a when the system is in belief state b. But the system should take action which will maximize the total expected reward in a dialogue which is based on the assumption that the belief state transitions are directed and depend on the previous value of b. The total expected reward is computed as

$$E(R) = E(\sum_{t=1}^{T} r(b_t, a_t)) \tag{3.3}$$

In POMDP, the belief state are probability distribution and hence continuous. Thus $p(b^{'}|b, a)$ denotes the probability density function. The expected future reward when starting in belief state b and following the dialogue policy π is recursively given by

$$V^{\pi}(b) = \sum_{a} \pi(b,a)r(b,a) + \sum_{a} \int_{b'} \pi(b,a)p(b^{'}|b,a)V^{\pi}(b^{'}) \tag{3.4}$$

When Working with Markov Decision Processes various other functions help in the task of policy optimization by choosing one of the policy(π) which maximizes $V^{\pi}(b_0)$ where b_0 is the start belief start e.g.

The Q-function $Q^{*}(b,a)$ is the expected future reward obtained by starting with a particular action and then following the policy and is given by

$$Q^{*}(b,a) = r(b,a) + \int_{b'} p(b^{'}|b,a)V^{\pi}(b^{'}) \tag{3.5}$$

The advantage function $A^{*}(b,a)$ is the difference between the Q-function and the total expected reward or value function V^{π} and is given by

$$A^{*}(b,a) = Q^{*}(b,a) - V^{\pi}(b) \tag{3.6}$$

The occupancy frequency $d^{*}(b)$ gives the expected number of times each state is visited. In the given equation $p(b_t = b$ denotes a probability density and $d^{\pi}(b)$ is a form of density function.

$$d^{\pi}(b) = \sum_{t=0}^{\infty} , p(b_t = b) \tag{3.7}$$

In a dialogue system, there are situations wherein it is clear to the system designer that taking a particular action is better than the others. *Summary actions* thus reduce the size of the action set during the policy learning and thus enable the system designer to embed the expert knowledge to allow learning to be quick and efficient. So along with machine actions $\overline{A}$ the summary actions A which are a subset of the machine actions are defined which are used for learning. Given a summary action a and a belief state b a mapping back to the original action set $F(a,b)$ is also defined. The use of summary actions is based on the summary POMDP idea proposed by [Williams and Young, 2005] which factors the state and actions according to a set of concepts.

3.8.1 Function approximation

The dialogue system can always reach a belief state which has never been observed earlier. So function approximation are to be used to generalize the past experiences to new belief states and depend on the actions as well as beliefs. The standard approach used is linear function approximation for either the value function V, the Q-function or the policy π in which the approximation is parameterised by a vector, θ, where the entries of θ are called *policy parameters*. The summary features which are computed from the belief states as a summary of the important characteristics are also used for function approximation and compiled into a vector $\phi(b)$. Features can include the entropy of the concept or the most likely probability for a concept. Given a set of features, different parameters will be used in the approximation, depending on which the action is taken. If the policy parameters for action a are denoted by θ_a then the approximation for the Q-function would be

$$Q(a, b, \theta) \approx \theta_a.\phi(b) \tag{3.8}$$

Some of the parameters from different actions are tied by the use of basis function $\phi_a(b)$ for optimization. Thus the approximation for the Q-function is given by

$$Q(b, a, \theta) \approx \theta.\phi_a(b) \tag{3.9}$$

3.9 Natural Actor Critic Algorithm

The algorithm Natural Actor Critic Algorithm [Peters and Schaal, 2008] is a modified form of gradient descent machine learning algorithm which we used in the framework and aided the spoken dialogue system to learn the parameters that optimize the expected future reward after the policy was parameterised using a suitable structure. Various other alternative algorithms e.g Temporal Difference Learning [Sutton and Barto, 1998] and Least Squares Temporal Difference Learning [Bradtke and Barto, 1996] have shown large fluctuations in policy performance during learning. The gradient descent algorithm uses the Euclidean matric as a measure of distance and iteratively subtracts a multiple of the gradient from the parameters being estimated. In general, the parameter space is known as Riemann space and for small changes in the parameters θ, a metric tensor, G_θ is defined such that the distance is $|d(\theta)|^2 = d\theta^T G_\theta d\theta$. [Amari, 1998] showed that for optimizing an arbitrary loss function in a general Riemann space the direction of the steepest descent also known as natural gradiend as compared the traditional vanilla descent is given by G_θ^{-1} , $\nabla_\theta L(\theta)$.[Amari, 1998] also showed the optimal metric tensor which gives distances that are invariable to scale with the parameters is the Fisher Information Matrix G_θ which is given by

$$(G_\theta)_{ij} = E(\frac{\partial\, logp(x|\theta)}{\partial\theta_i}\frac{\partial logp(x|\theta)}{\partial\theta_j}) \tag{3.10}$$

where $p(x|\theta)$ is a given probability distribuiton. [Peters and Schaal, 2008]showed that the Fisher Information Matrix for an Markov Decision Process is given by

$$G_\theta = \int_B d^\pi(b) \int_A \pi(a|b,\theta)\nabla_\theta\, log\pi(a|b,\theta)\nabla_\theta log\pi(a|b,\theta)^T\, da\, db. \tag{3.11}$$

The direction of the steepest descent is the inverse of this matrix multiplied by the vanilla gradient which is given by Policy Gradient Theorem as

$$\nabla_\theta V(b_0,\theta) = \int_B d^\pi(b) \int_A A^\pi(b,a)\pi(a|b,\theta)\nabla_\theta log\pi(a|b,\theta)\, da\, db \tag{3.12}$$

The equation depends on the advantage function and the occupancy frequency where the advantage function is approximated and the integral over the occupancy frequency is approximating using sampling methods where the rewards in the dialogue are grouped together to obtain suitable estimates.The sum of rewards gives an unbiased estimate of the sum of advantages and initial value function. If the approximate advantage function $\hat{A}_w(b,a)$ is chosen such that

$$\hat{A}_w(b,a) = \nabla_\theta log\pi(a|b,\theta)\,.\, w \tag{3.13}$$

where w minimises the average squared approximation error i.e.

$$\frac{\partial}{\partial w}\int_B d^\pi(b) \int_A \pi(a|b,\theta)(A^\theta(b,m)\ -\ \hat{A}_w(b,a)^2\, da\, db\ =\ 0 \tag{3.14}$$

then the required natural gradient is given by

$$G_\theta^{-1}\nabla_\theta V(b_0,\theta)\ =\ w. \tag{3.15}$$

The gradient has been used in the algorithm which is known as Natural Critic Algorithm that iterates between the evaluation step also known as critic step wherein the approximate advantage function is estimated and improving step wherein the actor improvement is done by changing the parameters by a multiple of natural gradient. The algorithm is sure to converge to a local maximum of the value function if the requirements are satisfied.

Algorithm 3.3 Natural Actor Critic Algorithm

```
for each dialogue, n do
    Execute the dialogue according to the current policy $\pi$
    Obtain the sequence of states $b_{n,t}$ and machine actions $a_{n,t}$
    Compute the statistics for the dialogue
        $\psi_n = \left[ \sum_{t=0}^{T_n} \nabla_\theta log\, \pi(a_{n,t}|b_{n,t}, \theta)^T, \, 1 \right]^T$
        $R_n = \sum_{t=0}^{T_n} r(b_{n,t}|a_{n,t})$
    Critic Evaluation
    Choose w to minimize $\sum_n (\psi_n^T w \; - \; R_n)^2$
    Actor Improvement
    Update the policy parameters
    $\theta_{n+1} \; = \; \theta_n \; + \; w_0 \, where \, w^T \; = \; [w_0^T, J]$.
    Propagate the impact and deweight the previous dialogue's
    $R_i \; \leftarrow \; \gamma R_i, \psi_i \; \leftarrow \; \gamma \psi_i$ for all $i \; \leq \; n$
end for
```

3.10 Evaluation

To evaluate the algorithm and the optimization of dialogues, a large number of human users dialogues are required which is prohibitive. Instead a simulator of the environment of the dialogue system is built and the system's policy can be optimized by interacting with the simulator instead of the human users. Optimum policies which are trained on the simulator are then used to bootstrap a policy which are further trained by interacting with the human user in the real world. The simulator includes both the user simulator, i.e. how the user behaves and responds when using the system and the error simulator i.e. how confusions are generated.

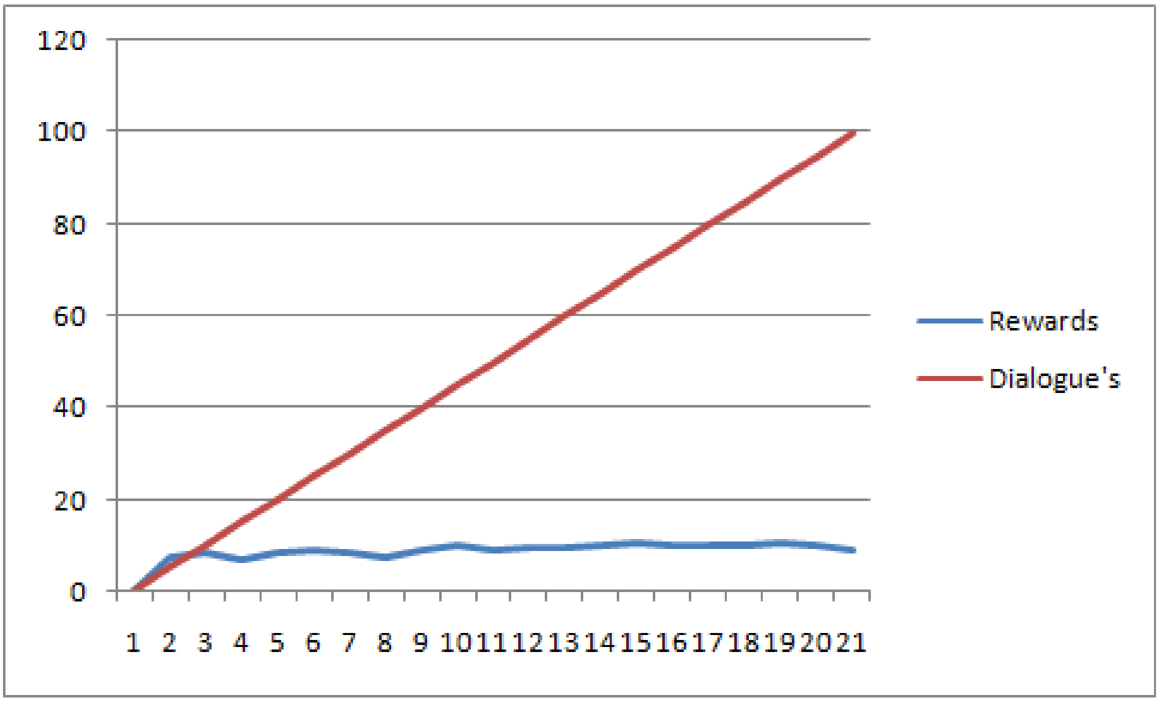


Figure 3.6: Plot showing the trend in the reward during policy training with a sample 100 dialogues -

3.11 Conclusion

The chapter has discussed the various methods to model and update the various states in a dialogue process. Bayesian Networks is an efficient and effective structure for modelling the various states in a spoken dialogue system which can be used by the dialogue manager for effective responding during the system's turn as its beliefs are updated with an approximated posteriori marginal distribution by the Loopy Belief Propagation algorithm which makes the system tractable by exploiting the conditional independence assumptions and limiting the time-slices for which the approximations are made in the factor graph. The chapter also shows how Natural Critic Algorithm can be used to learn the policy which tends to converge optimally. It has also been checked that this framework work well with human users as well as simulators.

CHAPTER 4

LANGUAGE MODELS

Spoken dialogue system has an uncertain parameter during the speech recognition which controls its performance that vary for the different users as well as for the same user during multiple repetitions of even the same dialogue. This chapter discusses how recognition errors in the users utterances can be handled by making use of language models. Language models can be improved over the stochastic language model for developing a syntactic structure based on word dependencies in local and non local domain The improved model copes with the issues of limited amount of training material and the exploitation of the linguistic constraints of the language.In this chapter we present how the proposed framework based on dynamic probabilistic model uses word dependencies based on their part of speech tags along with the tri-gram Model but also takes care of the influence of the word which are very far from the word being considered in a text and stores the word history in a dynamic cache for information mining using long distance dependency. The model based on second order Hidden Markov Model has been used and an improvement of 2% has been observed in the word error rate and 4% reduction in the perplexity when compared to the normal tri-gram model

4.1 Language Models

Language models captures the properties of a language and helps to predict a next word in the word sequence given the probabilities of the predecessor words which are calculated based on some given training text. The language model forms a very critical component for any spoken dialogue system as it defines the coverage and accuracy with which the system can understand what the user speaks and thus improving the performance of the dialogue manager. Statistical Language models also known as n-gram Language models characterize the word sequence as a Markov Process [Bahl et al., 1983] meaning the probability of a word given all previous words depends on the immediately preceding words. A n-gram is a sequence of n symbols (e.g words, syntactic categories etc) for some n $\geq$ 1. When n= 2 it is known as bi-gram language model i.e in a word sequence $w_1, w_2, ...w_i...w_n$ the word w_i is conditionally independent of the word history w_1, w_2, w_{i-2} given the preceding word w_{i-1}.

$$P(w_i|w_{i-1}, w_{i-2}, ..., 1) = P(w_i|w_{i-1}) \tag{4.1}$$

In this case the probability of the word sequence $Pw_1, w_2, ...w_i...w_n$ can be decomposed as the product of the conditional probabilities

$$P(w_1, w_2, ..., w_n) = \prod_{i=1}^{n} P(w_i|w_{i-1}) \tag{4.2}$$

Estimates of Probabilities in n-gram models are commonly based on maximum likelihood estimates i.e. by counting the words in the document on some given training text. The conditional context component also referred to as history can be extended to consider more than one word e.g trigram language model which is given by the following equation

$$P(w_1, w_2, ..., w_n) = \prod_{i=1}^{n} P(w_i|w_{i-1}, w_{i-2}) \tag{4.3}$$

The number of parameters in Markov Model is $|V|^n$ where V is the set of words and $|V|$ size of the vocabulary and the order of the Markov process is n-1. The Markov parameters are typically estimated using in-domain text and the problem of storage space and attaching a reasonable degree of confidence to the derived estimates are to be considered. In most of the research domains a vocabulary size of 65000 words and n=3 also referred as trigram language models have given successful results but the related used words outside this two word context are not taken into consideration which can lead to improvement in the perplexity of the model.

4.2 Historical background

The term language models originates from probabilistic models of language generation developed for automatic speech recognition systems in the early 1980s̀ [Jelinek, 1997]. Speech recognition systems use a language model to complement the results of the acoustic model which models the relation between words (or parts of words called phonemes) and the acoustic signal. The history of language models, however, goes back to beginning of the 20th century when Andrei Markov used language models (Markov models) to model letter sequences in works of Russian literature [Basharin et al., 2004]. Another famous application of language models are Claude Shannons̀ models of letter sequences and word sequences, which he used to illustrate the implications of coding and information theory [17]. In the 1990s̀ language models were applied as a general tool for several natural language processing applications, such as part-of-speech tagging, machine translation, and optical character recognition. Language models were applied to information retrieval by a number of research groups in the late 1990s̀ [4, 7, 14, 15]. They became rapidly popular in information retrieval research. By 2001, the ACM SIGIR conference had two separate sessions on language models containing 5 papers in total [13]. In 2003, a group of leading information retrieval researchers published a research roadmap ́Challenges in Information Retrieval and Language Modeling ́ [1], indicating that the future of information retrieval and the future of language modelling can not be seen apart from each other.

4.3 Statistical Language Modelling

Statistical Language Modeling is to build a statistical language model that can estimate the distribution of natural language as accurate as possible. A statistical language model (SLM) is a probability distribution P(s) over strings S that attempts to reflect how frequently a string S occurs as a sentence.By expressing various language phenomena in terms of simple parameters in a statistical model, SLMs provide an easy way to deal with complex natural language in computer science.The important application of SLMs is speech recognition, but SLMs also play a vital role in various other natural language applications as diverse as machine translation, part-of-speech tagging, intelligent input method and Text To Speech system.

4.4 Statistics Language Modelling techniques

1 **N-gram model and variants**

N-gram model is the most widely used SLM today. Without loss of generality we can express the probability $p(s)$ of a string s as

$p(s) = p(w_1)p(w_2|w_1)p(w_3|w_1w_2)...p(w_l|w_1...w_l - 1)$

4.5 Class based Semantic n-gram Model

Due to the sparse training text, we make use of Equivalence class based n-gram model where the probability of word is dependent on its history via the words semantic class [Solsona et al., 1993], groups of words that share a semantic category relevant to the spoken dialogue task. Considering some words as equivalent helps to reduce the word history equivalence classes to be modelled in the n-gram model. This is implemented by mapping a set of words to a word class by using a classification function. The domain knowledge can also be incorporated by classifying the relevant words into classes which may have some common feature e.g. In a medical assistance system, the user may select from a number of diseases which may be diagnosed based on a set of symptoms. In this case we first select a set of semantic classes $< diseases >, < symptoms >$ etc containing all the relevant diseases names and relevant symptoms appropriate for the domain and we then annotate the language model training corpus with the semantic classes: the training corpus is parsed with our natural language understanding grammar we find the constituents corresponding to the chosen semantic classes [Fosler-Lussier and Kuo, 2001]. And then compute the probability distributions e.g $P(w| < disease >)$ over all the words in the class. Consider a word sequence W $= w_1, w_2, ...w_i...w_n$ the word $C(w_i)$ and let be the class to which a word w_i belongs. The probability will be unique if the class are non overlapping else the probability $P(W)$ of the word sequence using a trigram semantic class model[Brown et al., 1992] is given by

$$P(w_1, w_2, ..., w_n) = \prod_{i=1}^{n} P(w_i|C(w_i))P(C(w_i)|C(w_{i-1}), C(w_{i-2})) \tag{4.4}$$

where $P(w_i|C(w_i))$ is the probability of the word w_i occurring in the semantic class $C(w_i)$. The probability distribution P(w—C(w)) depends on the semantic class. For instance, for the $< month >$ class we use the uniform distribution, but for the $< disease >$ class it is a function of the number of cases reported in the hospital

4.6 Proposed Model Description

Part of speech tagging is the act of assigning each word in a sentence a tag that describes how that word is used in the sentence. Typically, these tags indicate syntactic categories, such as noun or verb, and occasionally include additional feature information, such as number (singular or plural) and verb tense. Part of Speech Language Models have been used in speech recognition systems earlier [Dumouchel et al., 1988],[Jelinek et al., 1991] where the parameters are calculated using annotated training corpus. Cache language model uses a window of the ‘n‘ most recent words to determine the probability distribution of the next word. To achieve the dynamic behavior the recent

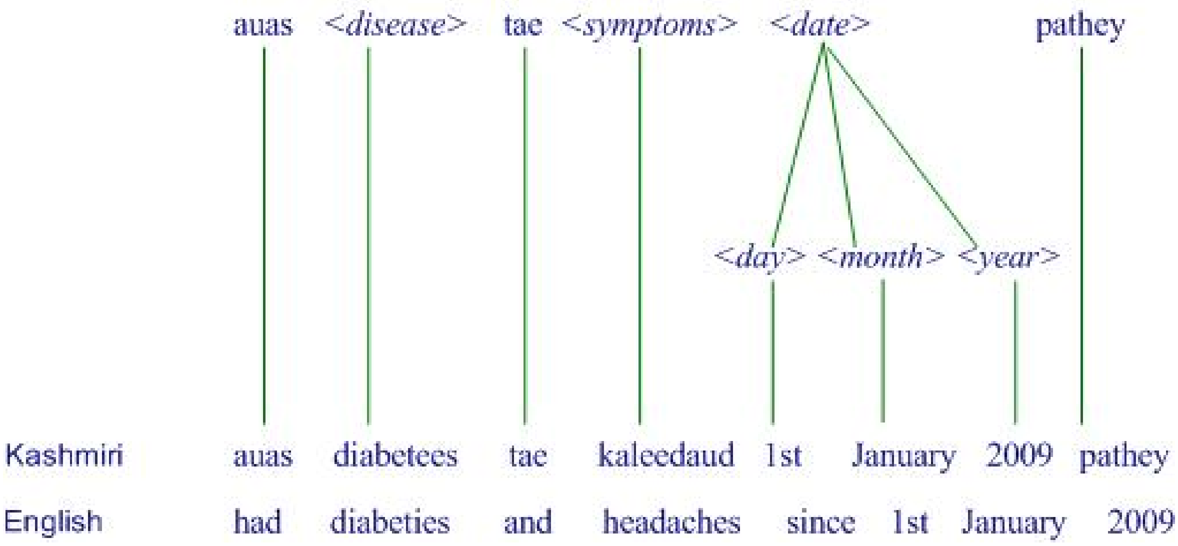


Figure 4.1: Semantic Parsing Example - The figure shows semantic parsing equivalent in Kashmiri and English

history has been stored and statistically evaluated in the caches earlier also [Kuhn, 1988],[Kupiec, 1989],. In the dynamic component,[Kuhn, 1988] used the a predicted POS in a trigram language model to adjust the probability of the next word. Each POS has a separate cache where the frequencies of all the word that occurred with a POS is used for the evaluation of the conditional probability of the next word. As a word is observed it is tagged and the appropriate POS cache is updated. The POS based Cache Semantic Model helps to identify the local dependencies between the words in a sequence based on the part of speech (POS) categories. The Parameters of POS model are of the form $P(w_i|S(w_i)) \times P(S(w_i)|S(w_{i-2}), S(w_{i-1}))$ which means that the POS category $S(w_i)$ is first determined for a word at position 'i' based on the POS category of the two words $S(w_{i-2}), S(w_{i-1})$ that precede it. First the various POS Categories are defined in the form of a vector which can be enhanced later. Then the model is to be trained for which a large training text corpus is required along with each words all possible POS categories that the word can take. Various words of the suitably sized training text are annotated with the unambiguous part of speech(POS) categories since many words can have multiple POS categories depending upon their role in the text. Estimates of the frequency of the words in the vocabulary for setting the initial probability in the model [Levinson et al., 1983]. Hidden Markov models (HMM) are stochastic models capable of statistical learning and classification. They have been applied in speech recognition and handwriting recognition because of their great adaptability and versatility in handling sequential signals [Thede and Harper, 1999]. So we use second order Hidden Markov models where the states correspond to POS categories and are labeled by the category they represent. The A matrix contains state transition probabilities, the B matrix contains output symbol distributions, and the C matrix contains unknown word distributions.The Parameters of POS model are of the form $P(w_i|S(w_i)) \times P(S(w_i)|S(w_{i-2}), S(w_{i-1}))$ which means that the POS category $S(w_i)$ is first determined for a word at position 'i' is based on the POS category of the two words $S(w_{i-2}), S(w_{i-1})$ that precede it. The probability of transitioning to a new state depends not only on the current state, but also on the previous state. This allows a more realistic context-dependence for the word tags than the first-order model. The elements of the output matrix have been assigned to word equivalence classes rather than the individual words which aid the estimation of the required number of parameters which is very large especially in different word types. Within these classes word have an uneven distribution and the transition matrix is set so that all the state transitions have an equal probability. The output matrix probability is based on the word occurrence probability $P(V_i)$ which is then converted to probabilities of the word equivalence classes $P(W_k)$. The probability of each equivalence class W_k is then divided equally among the POS categories that are in the equivalence class to give weights $F(W_k, C_i)$. This reflects the assumption that all words in an equivalence class can initially function equiv-probably as any POS category of the class. The output matrix elements for each state are constructed using the various $F(W_k, C_i)$. For each state, the elements are then normalized to sum to unity. The HMM model is then trained using Baum-Welch algorithm [Baum, 1972]. The algorithm (BW) is used for estimating the parameter values that maximize the likelihood of the training text belongs to a family of algorithms called Expectation Maximization (EM) algorithms. They all work by guessing initial parameter values, then estimating the likelihood of the data under the current parameters. These likelihoods can then be used to re-estimate the parameters, iteratively until a local maximum is reached. To determine the most likely state sequence Viterbi algorithm [Viterbi, 1967] has been used which maximizes the probability of seeing the test sentence.

The static language model has a probability distribution for the next word conditioned on the previous words which is obtained by taking mean over many documents. The static model has a problem that some words or word sequence are more likely to happen within a specific context can not depend on average over other documents. So to overcome this, we make use of a ”dynamic” model based on a word cache which contains frequency ordered linked list of words occurring in the previous text history. In a specific topic various words tend to be repeated as such there frequency count is incremented or if the word is not in the list, the list is updated with a initial count of 1. These counts are used to determine the conditional probabilities of words in the dynamic cache which participates in determining the correlation with the previous two words.

4.7 Experiment

We tested our model on a collection of test data sets (1) Academic speech for advising(The MICASE corpus) from University of Michigan and (2) The Trains corpus from University of Rochester were download and used for the study. The experiments were conducted to check the performance of Adaptive Hybid POS language model over tri-gram language model using word error rate (WER) and perplexity (PP) reduction as our measure.

4.8 Result and Discussion

Table 3.1 shows the perplexity of the static tri-gram language model and Adaptive Hybid POS language model over for the the test sets. The cache size was 1000 words and was updated word synchronously. The Adaptive Hybid POS language model yields from 8% to 12% reduction in perplexity, with the larger reduction occurring with the test sets with larger perplexity.

Test Data Set	Tri-gram Model	Adaptive Model
MICASE Corpus	95	90
TRAINS Corpus	167	126

Table 4.1: Perplexity of Static and Adaptive Model.

It was observed that an improvement of 2% has been observed in the word error rate and 4% reduction in the perplexity when compared to the normal tri-gram model.

Cache Size	**0**	**500**	**1000**
Perplexity	167	152	126

Table 4.2: Influence of Cache Size on perplexity of TRAINS Corpus

CHAPTER 5

SEMI-SUPERVISED LEARNING WITH HIDDEN STATE VECTOR MODEL

Spoken Language Understanding has been a challenge in the design of the spoken dialogue system where the intention of the speaker has to be identified from the words used in his utterances. Typically a spoken dialogue system comprises a four main components an automatic speech recognition system (ASR), Spoken language understanding component (SLU), Dialogue manager (DM) and an Speech synthesis system which converts the text to speech (TTS). Spoken Language understanding deals with understanding the intent from the words of the speakers utterances. The accuracy of the speech recognition system is questionable and researchers have provided various solutions to the problem and classifying the information may actually guide the dialogue manager in framing a response. Many models both statistical as well as empirical methods have been suggested for extracting information from text by automatically generating a language model after training from the annotated corpus.[Tur et al., 2005] When Statistical classifiers are used for classification they have to be trained using a large amount of task data which is usually transcribed and then assigned one or more predefined type to each utterance by humans, a very expensive and laborious process[Zhou et al., 2007]. But they do not perform well due to the lack of large scale richly annotated corpora. [Seymore et al., 1999] extracted the important information from the headers of computer science research papers by making use of Hidden Markov models. A statistical method based on HVS has been proposed to automatically extract information related to protein ? protein interactions from biomedical literature [Zhou et al., 2006]

5.0.1 Machine Learning

Learning as per the dictionary may be defined as ‘to gain knowledge or understanding of’ or ‘skill in by study,instruction or experience and ‘modification of a behavioural tendency by experience’. Herbert Simon defined Machine Learning as “Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more effectively the next time”. When a computer is programmed such that it optimizes its performance criterion P for a set of tasks T with respects to the changes in the environment E using example data or past experience. Computer Programs which can program themselves in different situations are then said to have learn from its previous experiences. Machine learning is used for

- Understanding and improving efficiency of human learning in complex systems. For example, In Computer-aided instruction we use machine learning to improve methods for teaching and tutoring people depending upon their pace and interest.
- Filling in skeletal or incomplete specifications about a complex large domain Artificial Intelligent system which cannot be completely derived by hand and require dynamic updating to incorporate new information. Learning new characteristics expands the domain or expertise and lessens the "brittleness" of the system.
- Discover new things or structure that is unknown to humans for example, Data mining. ...

Semi-supervised learning uses both supervised and unsupervised learning to learn from both annotated and unannotated sentences for classifications, clustering and so on. [Nigam et al., 2000] used Expectation-Maximization algorithm with a nave Bayes classifier on multiple mixture components for text classification. Small amount of labeled data is used to first build a model which is then used to annotate the instances of the unlabeled instances. The instance along with identified label which posses the more confidence measure are then added to the training set and participate in retraining of the model for the left out instances. The process is continued for the training of the remaining of the un-annotated sentences.

5.1 The Hidden Vector State Model

The basic hidden vector state model is a discrete Hidden Markov Model in which each HMM state represents the state of a push down automaton which encodes history in a fixed dimension stack. Each state consists of a stack where each element of the stack is a label chosen from a finite set of cardinality M+1,$C = \{c_1, .., c_M, c_{\neq}\}$. A HVS model state of depth D can be characterized by a vector of dimension D with most recently pushed element at index 1 and the oldest at index D. Each vector state is like a snapshot of the stack in the push-down automaton and transitions between states can be factored into a stack shift by n positions followed by a push of one or more new pre-terminal semantic concepts. The number of new concepts to be pushed is limited to one. The joint probability $P(W, C, N|\lambda)$ of a sequence of stack pop operations, word sequence W and concept vector sequence C is approximated as

$$P(W, C, N) = \prod_{t=1}^{T} P(n_t|W_1^{t-1}, C_1^{t-1}).P(C_t|W_1^{t-1}, n_t).P(C_t|W_1^{t-1}, n_t) \tag{5.1}$$

with the assumptions as

$$P(n_t|W_1^{t-1}, C_1^{t-1}) \approx P(n_t|c_{t-1})$$

$$P(C_t[1]|W_1^{t-1}, n_t) \approx P(c_t[1]|c_t[2?D_t])$$

$$P(C_t[1]|W_1^{t-1}, n_t) \approx P(w_i|c_i)$$

so we have

$$P(W, C, N) = \prod_{t=1}^{T} P(n_t|c_{t-1}).P(c_t[1]|c_t[2, \ldots, D_t]).P(w_i|c_i) \tag{5.2}$$

Where

(a) c_t denotes the vector state at word position t, which consists of D_t semantic concept labels (tags) i.e. $c_t = \{c_t[1], c_t[2], ..., c_t[D_t]\}$ where$c_t[1]$ is the preterminal root and $c_t[D_t]$ is the root concept normally represented by SS (Sentence Start)

(b) n_t is the vector stack shift operation and takes values in the range of $0 \ldots D_{t-1}$ where D_{t-1} is the stack size at word position $t-1$.

(c) $c_t[1] = c_{w_t}$ is the new preterminal semantic tag assigned to word w_t at word position t.

The key feature of the HVS model is its ability for representing hierarchical information in a constrained way which can be trained from only lightly annotated data. The generative process associated with HVS model consists of three steps for each position t :

(1) Choose a value for n_t.

(2) Select preterminal concept tag $c_t[1]$.

(3) Select a word w_t.

A set of domain specific lexical classes and abstract semantic annotations which limit the forward and backward search to include only those states which are consistent with these constraints for the model training must be provided for each sentence.

5.2 Semi-Supervised Learning

The main aim of the semi-supervised learning is to utilize the labelled utterances for annotating the unlabelled utterances in order to improve the performance of a classifier and reducing the human labelling effort. The semi-supervised learning technique used is as follows, Initially the human labeled task data is used to train the initial model which is used then to classify the unlabelled utterances. The machine labeled utterances whose confidence score value is above a threshold so that the noise due to classifier errors is reduced are added to the training data. If the input space is X and the output is $Y = \{-1, 1\}$ it is known as binary classification. . Suppose E_L is the small set of labeled sentences $E_L = \{< s_1, a_1 >, < s_2, a_2 >, ..., < s_i, a_i >\}$ where $S = \{s_1, s_2, .., s_i\}$ is the set of sentences and $A = \{a_1, a_2, .., a_i\}$ is the set of corresponding annotation for each sentence. And E_U is the large set of unlabelled data $E_U = \{s_{i+1}, s_{i+2}, ?..s_{i+u}\}$. The process of predicting the labels A_U of the unlabelled data S_U is known as the transduction. The process of constructing a classifier $f : X = \{-1, 1\}$ on the whole input space using the unlabeled data comes under the purview of semi-supervised learning

5.3 Related Work

In Language Processing framework there are two approaches viz certainty based approaches and committee based approaches of having control over the type of inputs on which it trains.In certainty based approaches, a small set of annotated examples is used to train the system, the system then labels the unannotated sentences and determines the confidence for each of its prediction. The sentences with lower confidence are then presented to the labelers for annotation. In Committee based methods, a small set of annotated sentences are used to create a disjoint set of classifiers, which are then used to classify the unannotated sentences. The sentences where the classification differ much are manually annotated. [Nigam et al., 2000]learned from both labeled and unlabelled data based on combination of Expectation Maximization and a Nave Bayes classifier on multiple mixture components per class for task of text classification. [Yarowsky, 1995] used self training for word sense disambiguation. Rosenberg et al [Rosenberg et al., 2005] applied self training to object detection from images. Self training builds a model based on the small amount of labeled data and then uses the model to label instances in the unlabeled data. The most confident instances together with their labels participate in the training set to retrain the model. [Ghani, 2002] proposed an algorithm for exploiting the labeled as well as un- labeled data using the co training with Expectation Maximization(CO-EM). [Allen et al., 2002] used semi-supervised learning for automation speech recognition and have shown improvements for statistical language modeling where they exploited confidence scores for words and utterances computed from ASR word lattices.

5.4 Proposed Framework

A probabilistic framework is used to describe the nature of sentences and their annotations where semantic annotations are considered as the class label $g \in G$ for each sentence with the following two assumptions

(a) If $|G|$ is the number of distinct annotations in the labeled setE_L where$E_L = \{(s_1, a_1), (s_2, a_2), ?., (s_L, a_L)\}$ then the data are produced by $|G|$ is the number of distinct annotations in the labeled set probability models.

(b) There is a one to one correspondence between probability components and classes. Considering the each individual annotation as a class, the likelihood of a sentences_i is given by

$$P(s_i|\lambda) = P(a_i = g_j|\lambda)P(s_i|a_i = g_j, \lambda) \tag{5.3}$$

Where g_j is the annotation of the sentence s_i and λ represents the complete set of HVS model parameters. Since the domain of possible training examples is $s_{|L|+|U|}$ and the binary indicators are known for the sentences in E_L and unknown for the sentences in E_U. The class labels of the sentences are represented as the matrix of binary indicators Z where

$$Z_{ij} = \{ +1 \quad if a_i = g_j 0 \quad otherwise$$

Then we have

$$P(s_i|\lambda) = \sum_{j=1}^{|G|} z_{ij} P(g_j|\lambda) P(s_i|g_j, \lambda) \tag{5.4}$$

Calculating the maximum likelihood estimate of the parameters λ i.e. $argmax_{\lambda} P(W, C, N|\lambda)$ for learning the HVS model. The annotation A for the word sequence W can be determined by $\{C, N\}$ i.e the concept vector sequence C and the series of stack shift operations N and $\{C, N\}$ can be inferred from A.Thus $argmax_{\lambda} P(W, C, N|\lambda)$ can be rewritten as $argmax_{\lambda} P(W, A|\lambda)$ which can further be rewritten as $argmax_{\lambda} P(E|\lambda)$ which is the product over all the sentences assuming each sentence is independent of each other. The probability of the data is given by

$$P(E|\lambda, Z) = \prod_{S_i \in E} \sum_{j=1}^{|G|} Z_{ij} P(g_j|\lambda) P(s_i|g_j, \lambda) \tag{5.5}$$

The complete log likelihood of the parameters $l_g(E|\lambda, Z)$ can be expressed as

$$l_g(E|\lambda, Z) = \sum_{s_i \in E} \sum_{j=1}^{|G|} Z_{ij} log P(g_j|\lambda) P(s_i|g_j), \lambda) \tag{5.6}$$

To improve the performance of classifier, the methods used are based on classification and Expectation Maximization. Both the methods assume that there is some training data available for the initial classifier. The main aim is to use this classifier to label the unlabelled data automatically and to then improve the classifier performance using machine labeled utterances. Semi-supervised learning based on classification measures the edit distance between the POS tag sequences of the sentences in E_L and POS tag sequences of sentences in E_U to automatically generate the annotation for the unlabelled sentences. The edit distance or Levenshtein distance of two strings, s1 and s2, is defined as the minimum number of point mutations required to change s1 into s2, where a point mutation can be either changing a letter or inserting a letter or deleting a letter.If X and Y are two pos tag sequences of length n and m respectively, a tabular computation which contains the score of the optimal alignment between the initial segment from X and the initial segment from Y is calculated using the following algorithm.

Algorithm 5.1 Minimum Edit Distance Algorithm

```
**Intialize :** Set $D(i,0) = 0$ and $D(0,j) = 0$.
for each $i$ from 1 to m do
  for each $j$ from 1 to n do
    $D(i,j) = D(i-1,j) + 1$
    $D(i,j) = D(i,j-1) + 1$
    if $X(i)$= $Y(j)$ then
      $D(i,j) = D(i-1,j-1)$
    else
      $D(i,j) = D(i-1,j-1) + 2$
    end if
  end for
end for
```

Dynamic programming which solves problems by combining solutions to sub problems is used comprising of edit distance matrix $D(i, j)$. By this technique we first calculate $D(i, j)$ for smaller $i, j$and compute larger $D(i, j)$based on the previous computed smaller values i.e compute $D(i, j)$ for all $0 < i < n$ and $0 < j < m$. Given two sentences S_i, S_j and their corresponding POS tag sequences $T_i = a_1a_2..a_{ni}$ and $T_j = a_1a_2..a_{nj}$, the distance between the two sentences is defined as $Dist(S_i, S_j) = -D(n_i, n_j)$ where $D(n_i, n_j)$ is the distance measure of optimal alignment between two POS tag sequences T_i and T_j.

5.5 Distance-Weighted Nearest Neighbour Algorithm

Classification a spoken dialogue learning uses a finite number of labeled examples and selects a hypothesis is expected to generate few errors on the future examples. In case of spoken dialogue systems human labeling of the the spoken utterances has a wide impact on the quality of the machine labeling of the unlabeled sentences. The basic elements to handle by classification algorithm are word lattices which may contain a single word or a collection of words with some weight or probability [Blum and Mitchell, 1998]. The technique which we have used for classification is Distance-Weighted Nearest Neighbour Algorithm. Since the training Input variables consists of the set ¡X,Y¿ where X contains represents the word and Y represents its semantic annotation, the algorithm finds the training points which have the closest edit distance to the queried word. It assigns weights to the neighbours based on their d̀istancèfrom the query point, the Weight are inverse square of the distances. and then classifies according to the mean value of the k̀nearest training examples. All the training points influence a particular instance.

5.6 Semi-supervised Learning based on expectation maximization

The EM algorithm is an efficient iterative procedure to compute the Maximum likelihood (ML) estimate in the presence of missing or hidden data. In ML estimation, we wish to estimate the model parameter(s) for which the observed data are the most likely. So we cluster the sentences in E_L and E_U. The original model will contain more sentences since some sentences in E_U will have the similar semantic structure with those sentences in E_L which have been used to train the HVS Model but adding should be based on some confidence measure so that the performance of the model is improved. To do this a parameter DG_f which represents the degree of fitness is to be used for selecting the sentences based on parsing information I_p, structural information I_s and complexity information I_c[2]. These parameters of a sentence are defined as
Parsing information I_p describes the information in the parsing result and is defined as

$$I_p = 1 - \frac{\sum_{j=1}^{n} KEYI(S_{ij})}{\sum_{j=1}^{n} KEY(S_{ij})}$$

Where ǹ d̀enotes the length of the sentence s_i, s_{ij}denotes the j^{th} word of the sentence s_i and the functions $KEYI(s_{ij})$ is equal to 1 if $s_(ij)$is a word in the E_L and 0 otherwise. $KEY(s_{ij})$ is 1 if $KEYI(s_{ij})$ is 1 and the semantic tag of s_{ij} is not known and 0 otherwise.
Structure information I_s is a measure of similarity between the structure information of a sentence s_i and the sentences s_j in E_L which is given by

$$I_s = 1 - \frac{\min ?(Dist(s_i, s_j))}{\max\left(Dist(s_k, s_j)\right)} + \frac{NUM(C(s_i))}{|E_L|}$$

Where $s_j \in E_L$ and $s_k \in E_U$, $C(s_i)$ denotes the cluster where $s_i$is located, $Dist(s_i, s_j)$ is the edit distance measure between sentence s_i and s_j. $NUM(C(s_i))$ is the number of sentences in the cluster $C(s_i)$.

Complexity information I_c is based on the length of the sentence s_i and the max length of the sentence s_j where $s_j \in E_L \cup E_U$. $I_c$is given by

$$I_c = 1 - \frac{length(s_i}{\max?(length(s_j)|s_j \in E_L \cup E_U)}$$

Since the measure of selecting a sentence is based on the degree of fitness DG_f which is given by

$$DG_f = \beta_p I_p + \beta_s I_s + \beta_c I_c + \beta_o$$

The coefficients $\beta = (\beta_p, \beta_s, \beta_c, \beta_o)$ are calculated using the method of least squares and β is selected to minimize the residual sum of squares.

$$RSS(\beta) = \sum_{i=1}^{N} (DG_f^{'} - DG_f)^2$$

The parameter β is estimated from the ǹ s̀et of training data, $DG_f^{'}$ is the estimated value and DG_f is the observed value. First a sample corpus of words are identified from the travel domain. Then a semantic tag based on the class is attached for identifying interactions. The vertibi decoding algorithm is used to parse the sentences of the E_L. For the sentences in E_U selection is done based on the parameters i.e. DG_f. Thus the sentences in E_U would be added to the set of sentences with annotation and participate in further automatically annotating sentences in E_U.

5.7 Experimental Evaluation

.

5.7.1 Methodology

To evaluate the proposed model the training data was split into two data sets corpus I comprising of 200 sentences out of which 100 sentences with manual annotation from travel domain are added toE_L for training the HVS model and 100 sentences were added to E_U. First clusters are created from the learned sentences based on the edit distance measure and then semi supervised learning based on expectation maximization was applied to the sentences in E_U The corpus II comprised of the 250 sentences which incremented the 200 sentences by 50 more sentences with annotation for learning the

HVS model. And then out of 100 sentences 47 sentences were semantic annotated successfully with out any human labeling by the algorithm.

5.7.2 Results

. The experimental results for the baseline HVS model trained on sentences in E_L contained 74 classes when classification was performed. 8 Experiments were performed for subset of sentences in E_U with the k = 1,2,3 based on Distance-Weighted Nearest Algorithm. The overall precision was calculated by

$$OP = \frac{NS_c}{NS_c + NS_{ic}}$$

where

- Number of sentences for which annotation was done correctly
- Number. of sentences for which annotation was done Incorrectly

based on classification. The overall precision in the travel domain data set was observed at 65.4% with k=3 when only sentences from E_L were used. The HVS Model was incrementally trained with these newly added sentences from E_U based on the sentence selection based on expectation maximization which improved the performance by 4.6%

Experiment	**Precision** %(E_L)	**Precision** %$(E_L + E_U)$ **?**
1	54.3	62.1
2	58.7	59.6
3	59.9	61.7
4	64.1	65.4
5	65.7	59.2
6	57.1	68.7
4	58.2	65.8
5	52.3	67.3

Table 5.1: Precision %age of Semi-Supervised Learning on Labeled and using labeled training data for unlabeled data set

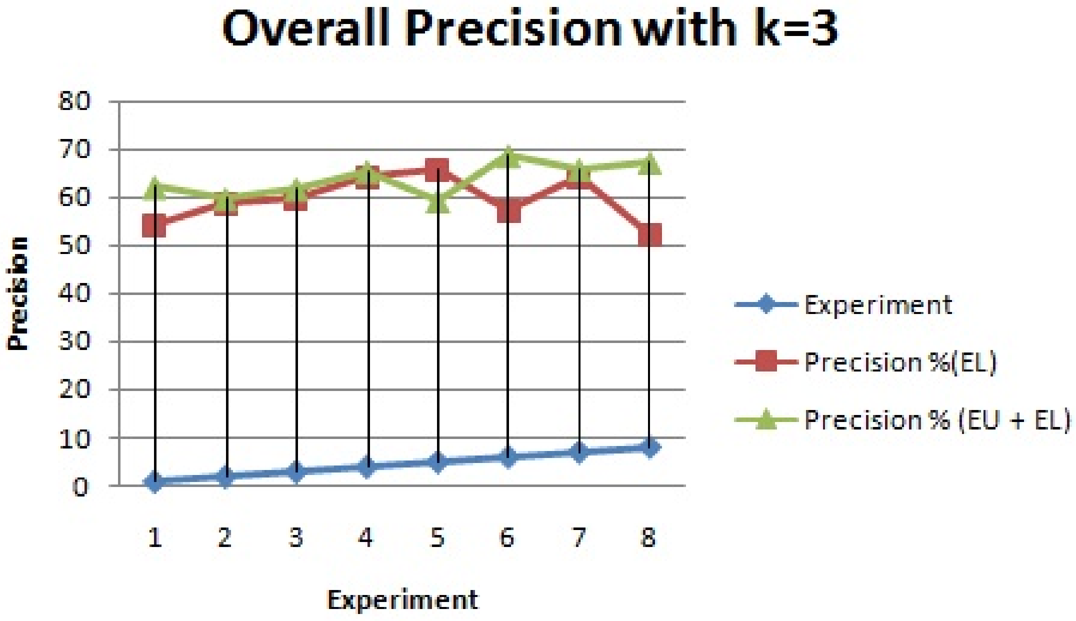


Figure 5.1: Semantic Parsing Example - The figure shows semantic parsing equivalent in Kashmiri and English

5.8 Conclusion

In this chapter we have used two semi-supervised learning techniques which have made use of both labeled and unlabeled data to improve the performance of the HVS model. The overall performance was improved by nearly 4-5%. In future we will use the machine learning technique like SVM or Kernels for dealing with problems where minimum labeled data is available.

CHAPTER 6

PROSODIC LEARNING

Utterances used by a human while framing a response during the interaction with a software agent like spoken dialogue system(SDS) has valuable information as regards internal mental state of the user is concerned. Sentiment analysis is an analysis of the mental state of a person, his opinion, appraisal or emotion towards an event, entities or their attributes. The users level of certainty about a topic could be determined by the analysis not only of the text used in the utterance but by studying the prosody information structure. Prosody reveals Information about the context by highlighting information structure and aspects of the speaker hearer relationship. Most often it is observed that the speakers internal state is not depicted by the words he uses but by the tone of his utterance or facial expression of the user.

The dialogue cycle for more general human-computer interactions include information other than speech which need some method for its identification and then using these non-speech inputs so that they can included by adding inputs to and outputs from the dialogue manager. This does not break the dialogue cycle structure since the dialogue manager can extend its definition of state to include this extra information. Similarly, the output from the dialogue manager can be extended to include more than a dialogue act and could include other actions as well. [Bohus and Horvitz, 2009] provides an example of a spoken dialogue system with multiple parties and visual sensory inputs. In this chapter we had analyzed a sample of student conversations after a lecture on operating system subject and based on prosodic features during few questions determine whether they were certain, uncertain or neutral about the lecture contents. This paper uses PRATT a tool for speech analysis which uses 15 acoustic features to determine the certainty of the responses of the user through classification by RAPIDMINER based on the prosody information which will actually aid the dialogue management component of the Spoken Dialogue System in framing a better dialogue strategy.

6.1 Introduction

Spoken language is an intuitive form of interaction between humans and computers. Spoken Language Understand has been a challenge in the design of the spoken dialogue system where the intention of the speaker has to be identified from the words used in his utterances. Typically a spoken dialogue system comprises a four main components an automatic speech recognition system (ASR),Spoken language understanding component (SLU), Dialogue manager (DM) and an Speech synthesis system which converts the text to speech (TTS). Spoken Language understanding deals with understanding the intent from the words of the speakers utterances. The accuracy of the speech recognition system is questionable and researchers have provided various solutions to the problem of automatic speech recognition which lagged behind human performance [Baker et al., 2009] there have been some notable recent advances in discriminative training [He et al., 2008] e.g., maximum mutual information (MMI) estimation [Kapadia et al., 1993], minimum classification error (MCE) training [Juang et al., 1997], [McDermott et al., 2007], and minimum phone error (MPE) training [Povey and Woodland, 2002], [Povey, 2004]), in large-margin techniques (such as large margin estimation [Jiang and Li, 2007] large margin hidden Markov model (HMM) [Sha and Saul, 2006], large-margin MCE [Yu et al., 2006], and boosted MMI [Povey et al., 2008], as well as in novel acoustic models (such as conditional random fields (CRFs) [Hifny and Renals, 2009], hidden CRFs [Gunawardana et al., 2005] [Yu and Deng, 2010] and segmental CRFs [Zweig and Nguyen, 2010],training densely connected, directed belief nets with many hidden layers which learn a hierarchy of nonlinear feature detectors that can capture complex statistical patterns in data [Hinton et al., 2006]. There are many cases of experiences by the users when the computers either do not understand the intended meaning of the user even after correctly recognizing the spoken utterances. One of the reason may be that in a face to face human conversation, there are contextual, audio and visual cues [Krahmer and Swerts, 2005] which aid the knowledge requirements of the users for the efficient communication as the users other than the contextual are able to sense the mood and tone of the user by which they come to know whether the speaker is certain or not. This is, absent in a dialogue between a computer and a human because in many potential applications there is only audio input and no video input. If the Spoken Dialogue Systems are improved to use the prosodic information from the spoken utterance they will definitely benefit from the level of certainty of the user [Heather et al., 2011] such as spoken tutorial dialogue systems [Forbes-Riley and Litman, 2009], language learning systems [Alwan et al., 2007] and voice search applications [Paek and Ju, 2008]. Our primary goal is to make use of prosodic information for aiding the dialogue manager in selecting the dialogue strategy for effective interaction and influencing the final outcome. Technically Prosody is defined as the rhythm, stress, and intonation of speech which reflect various features such as emotional state of the speaker, the form of the utterance (statement, question, or command, the presence of irony or sarcasm, ; emphasis, contrast, and focus or other elements of language that may not be encoded by grammar or choice of vocabulary Prosodic information of an utterance can be used to determine how certain a speaker is and hence the internal state of mind [Lee and Narayanan, 2005] which can be used for tasks from detecting frustration [Ang et al., 2002], to detecting flirtation [Ranganath et al., 2009] and other intentions. The model proposed that uses prosodic information to classify utterances has effectively coloured the system responses in a travel based information system and performed better than a trivial non-prosodic baseline model. In the context of human computer interaction, the study of prosodic information has been aimed at extracting mood features in order to be able to dynamically adapt a dialog strategy by the automatic SDS.

6.2 Corpus and Certainty annotation

It is very important to understand that not only what words are spoken by a speaker in his utterance but how the words are spoken along with the certainty factor can actually guide the dialogue process between the machine and the user. The spoken utterance may be perceived as uncertain, certain, neutral or mixed which helps the dialogue system to make a guess about the mental state of the user about the utterance or about the concept about which he is speaking about. In this paper we examine impact of a lecture and certainty of students as it is expressed within the context of a spoken dialogue.

AGENT : What is an operating system.

STUDENT : It is a set of software or hardware may be (UNCERTAIN).

AGENT : Is it hardware or software.

STUDENT :software(CERTAIN)

AGENT : What do you know about round robin scheduling.

STUDENT : Uh-uhh (NEUTRAL)

A corpus of 15 lecture related dialogues are selected and after listening each sentence of the student is labeled by an annotator with either certain or uncertain or neutral. The dialog were also lexically annotated based on the words used as certain, uncertain and neutral. The percentage of sentences with certainty, uncertainty and neutral for the auditory and lexical conditions are shown in the table 6.1.

Condition	**Certain**	**Uncertain**	**Neutral**	?
Auditory	22.3	18.4	59.3	
Lexical	12.1	11.7	76.2	

Table 6.1: Percentage of corpus with different levels of certainty, annotated by listening to the audio of the dialogue context and annotated based on the lexical structure of the dialogues.

It was observed that 40.7% non-neutral corpus could be decided as certain or uncertain based on the audio and the dialog context compared to the 23.8 % based on the lexical information. As such we used the acoustic-prosody features for further information about the certainty or uncertainty.

6.3 Prosodic Model

For the basic model we compute values for 15 prosodic features as given in the table 6.2 for each utterance in the corpus of the student lecture set using PRATT (a program for speech analysis and synthesis)[Boersma, 2002] and wavesurfer for extracting the f0 contour. Feature values are represented as zscores normalized by speaker. The temporal features like voice breaks, unvoiced frames, degree of voice breaks, Total duration are not normalized.

The set of features were selected in order to be comparable with Liscombe et al [Liscombe et al., 2005] who used the same features along with turn related features for classifying uncertainty. The set of features were selected in order to be comparable with Liscombe et al [31] who used the same features along with turn related features for classifying uncertainty

Number of Features	**Features** ?
6	Minimum, Maximum and Standard Deviation, Relative position Min f0, Relative position Maxf0, Fundamental frequency (f0)(Statistics of Pitch)
4	Minimum, Maximum Mean and Standard Deviation (RMS),(Statistics of Intensity)
1	Ratio of voiced frames to total frames in the speech signal as an approximation of speaking rate
2	Total silence, Percent silence
2	Speaking duration, Total duration.

Table 6.2: Extracted and selected features.

6.4 Classification Results

The features extracted are used as input variables to RAPID MINER machine learning software which built C4.5 decision tree models which iteratively builds weak models and combines them to form a better model to predict the classification of unseen data. As an initial model we train a single decision tree using the selected 15 features as listed in Table 6.2. The model was evaluated over all the utterances of the corpus and it classified within the classification classes, certain, uncertain and neutral and cross validated with an accuracy of 65% as compared to the non-prosodic model which had a an classification accuracy of 51.1%.

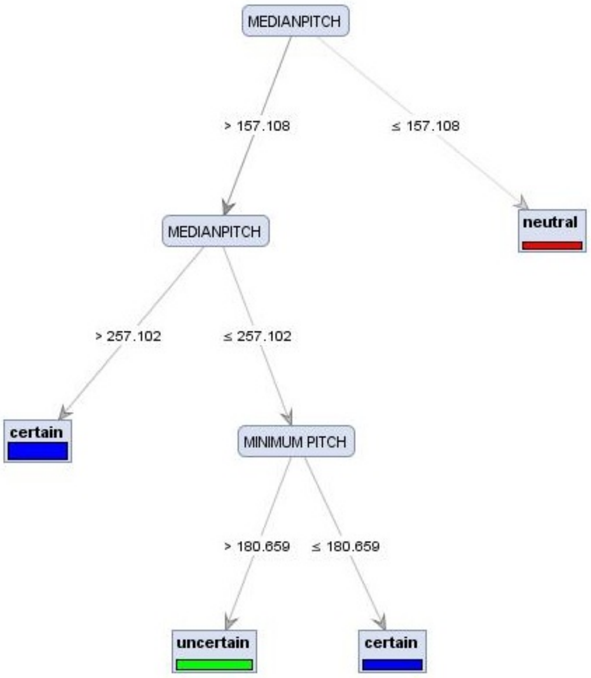


Figure 6.1: Undiscretized Data Decision Tree

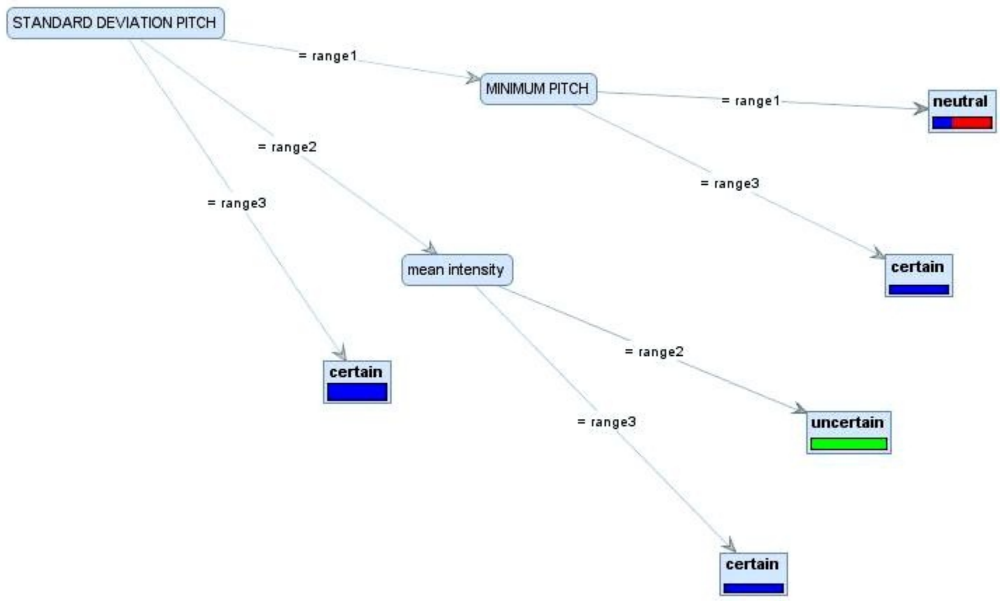


Figure 6.2: Decision tree obtained after data discretisation

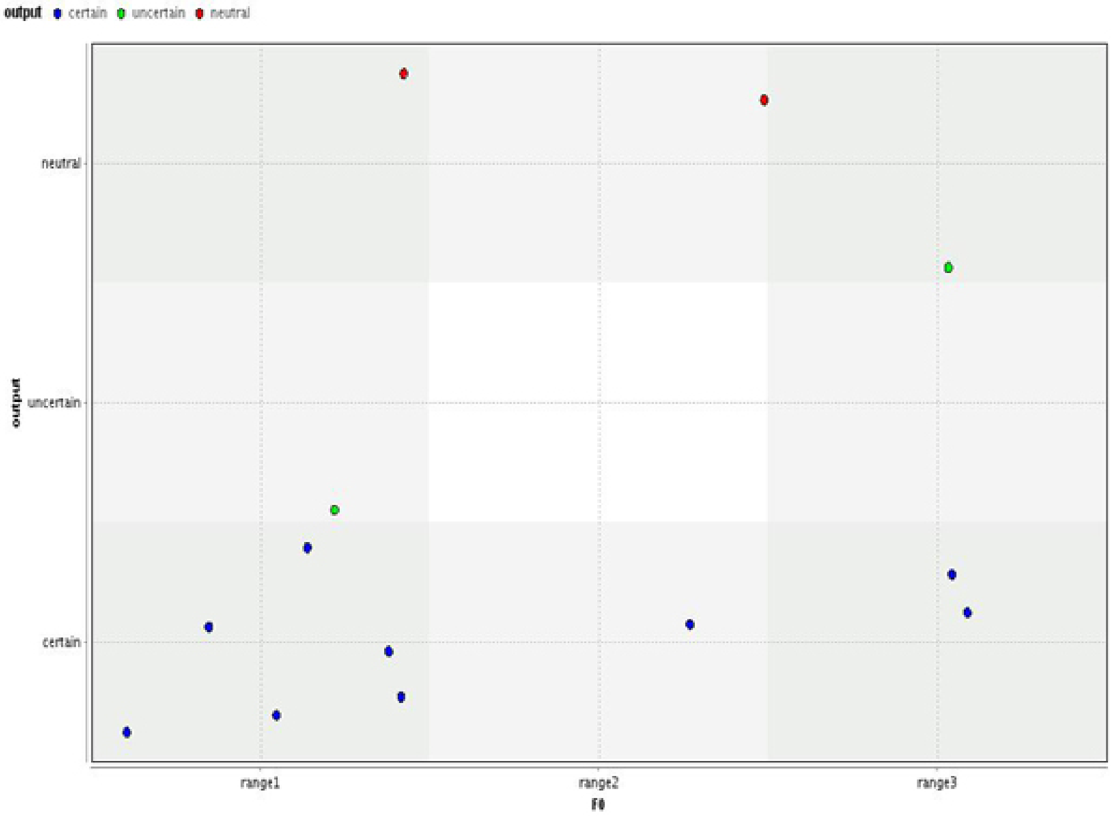


Figure 6.3: Figure indicating the Pitch versus the output of different samples

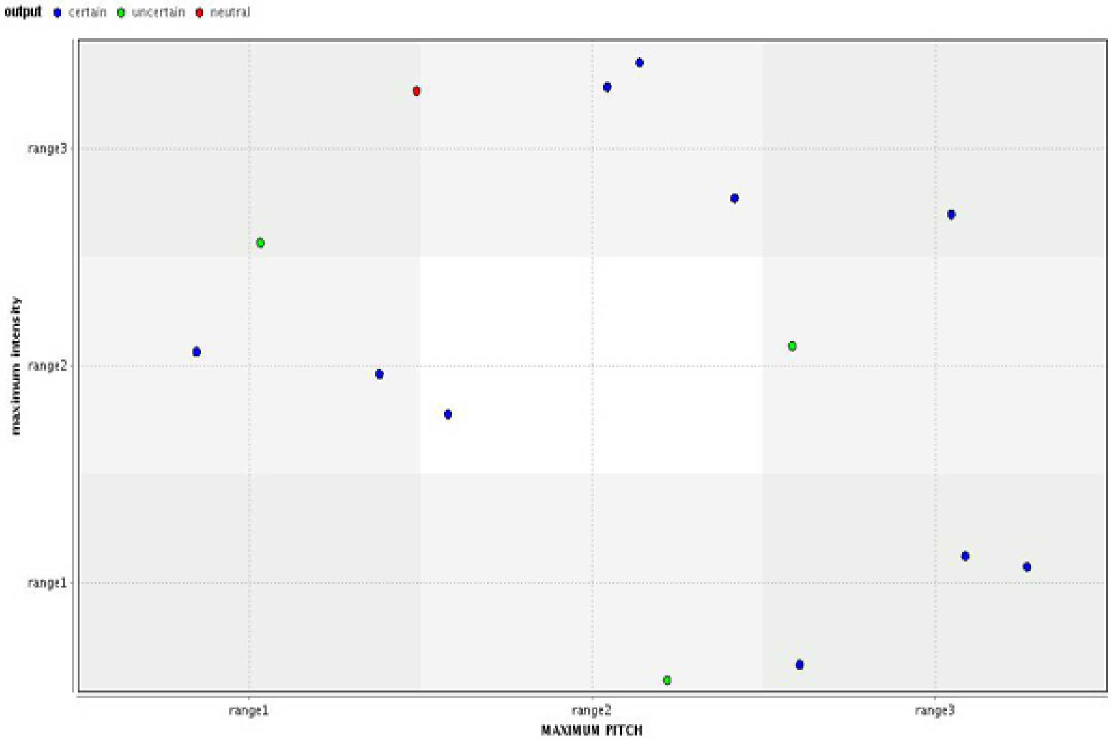
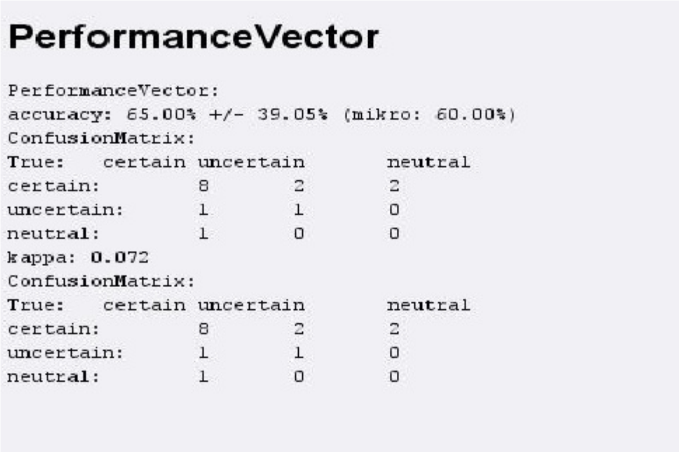


Figure 6.4: Figure indicating the maximum intensity versus the maximum pitch of different samples



PerformanceVector

```
PerformanceVector:
accuracy: 65.00% +/- 39.05% (mikro: 60.00%)
ConfusionMatrix:
True:    certain uncertain       neutral
certain:         8       2       2
uncertain:       1       1       0
neutral:         1       0       0
kappa: 0.072
ConfusionMatrix:
True:    certain uncertain       neutral
certain:         8       2       2
uncertain:       1       1       0
neutral:         1       0       0
```

Figure 6.5: Performance Statistics

CHAPTER 7
CONCLUSIONS & FUTURE WORK

7.1 Conclusions Drawn

This thesis has shown that the representing the dialogue using Bayesian approaches provides an efficient and effective solution for handling the inherent uncertainty and also enable the dialogue history to be taken into consideration when the design of the dialogue manager is to be considered. Various algorithms both for exact as well as for approximate updates for highly complex real-world systems were developed and tested on the sample. The parameters for the belief updating models can be learned on data that is not annotated for dialogue state, which makes the application of such models for new tasks relatively simple. Using machine learning techniques, the system can learn optimal policies and use that for its strategy formulation for a given task.These policies don't require any human intervention, since all policy decisions are learned automatically. This also simplifies the design process of a dialogue system for new domains.

Experiments have shown that the various approaches presented in this thesis outperform traditional approaches. The proposed framework which includes Semi-Supervised learning of Hidden Vector State model and utilizing prosodic information results in improved handling of uncertainty, improved re-scoring ability of the user model, improved decision making of the policy and finally improved overall performance of the dialogue system. In summary, the approaches described here have several key advantages when compared to alternative approaches. When compared to traditional hand-crafted approaches there are following advantages:

- The system becomes flexible to handle noise.
- The system learns the policies using the reinforcement learning techniques automatically and as such relieves the system designer from additional effort.
- Belief updates are done using possible efficient strategy based on classes and hence happen to computationally efficient.
- Enables building complex spoken dialogue systems.
- More flexibility is allowed e.g User goal can change during the dialogue process.
- Parameters can be learned without annotations of dialogue state

Chapter 3 is an exploratory work which identifies current issues in spoken dialogue representation. Thereafter, we propose a framework which aim at mitigating the current issues in dialogue representation and an efficient inferencing. In addition to this, a set of factors are also presented which can help us in choosing appropriate learning technique which aids in automatically policy formulation for the dialogue manager.

Chapter 4 presents how Language Modelling using adaptive hybrid POS cache model can help in confidence scoring for handling errors during the automatic spoken recognition.The model proposed using a window based tri-gram language model which is capable or re utilizing the information for better scoring in future to correct the wrongly identified words by the ASR. The perplexity as well as Word Error Rate had improved over the traditional approaches

Chapter 5 presents an empirical study of how semi-supervised learning using Hidden Vector State Model can also aid in effective learning.We have used two semi-supervised learning techniques which have made use of both labeled and unlabeled data to improve the performance of the HVS model. The overall performance was improved by nearly 4-5%.

Chapter 6 presents an empricial study to show that prosodic information can be used to identify the intention as well as the mental structure which is normally not given by the words that the speaker speaks and is used in human to human conversations for responding. The results show that it can be determined that whether the speaker is certain, uncertain or neutral about a subject based on the evaluation of prosodic information, like frequency, pitch or intensity etc.

7.2 Future Work

There are some limitations in the present research that stem from the particular research focus and scope that have been chosen. With so many modelling techniques and the very inadequate quantitative and qualitative knowledge about them, we strongly believe that there is a need of much more research and evidence in the spoken dialogue systems. The work in this thesis can be extended or replicated further to produce more realistic, generalized and implementable results.

The work in the thesis can be extended in the following ways:

1. The system design has to have the knowledge of the domain for which the spoken dialogue system is to be designed and has to embed the knowledge using ontologies about the concepts. Future work will need this to be done automatically by discovering the structure of the problem using machine learning.

2. To evaluate the policy learning it has to be tested with a dialogue simulator. Current simulation techniques require large amounts of development time, which makes changing the domain of a system difficult. Future work will develop methods for automatic learning of the system's user model which could then be extended to user simulation and also evaluate the simulation.

3. To use the non-verbal cues like gestures for framing the dialogue strategy in addition to prosodic information and use kernel methods for this process

4. Refining the proposed framework, if required.

End

REFERENCES

Allen, J. Natural language understanding. 1987. 11

Allen, J., Hunnicutt, M., and Klatt, D. From text to speech: The mitalk system. *216 pp*, 1987. 14

Allen, L., Abella, A., Alonso, T., and Jeremy, H. Automated natural spoken dialog. 2002. 70

Alwan, A., Bai, Y., Black, M., Casey, L., Gerosa, M., Heritage, M., Iseli, M., Jones, B., Kazemzadeh, A., Lee, S., et al. A system for technology based assessment of language and literacy in young children: the role of multiple information sources. In *Multimedia Signal Processing, 2007. MMSP 2007. IEEE 9th Workshop on*, pages 26–30. IEEE, 2007. 79

Amari, S. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998. 51

Ang, J., Dhillon, R., Krupski, A., Shriberg, E., and Stolcke, A. Prosody-based automatic detection of annoyance and frustration in human-computer dialog. In *Seventh International Conference on Spoken Language Processing*, 2002. 80

Austin, J. *J.L. (1962). How to do things with words.* Oxford University Press, New York. 9, 1962. 22

Bahl, L., Jelinek, F., and Mercer, R. A maximum likelihood approach to continuous speech recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):179–190, 1983. 57

Baker, J., Deng, L., Glass, J., Khudanpur, S., Lee, C., Morgan, N., and O'Shaughnessy, D. Developments and directions in speech recognition and understanding, part 1. *Signal Processing Magazine, IEEE*, 26(3):75–80, 2009. 79

Basharin, G., Langville, A., and Naumov, V. The life and work of aa markov. *Linear Algebra and its Applications*, 386:3–26, 2004. 58

Baum, L. An equality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3:1–8, 1972. 62

Bellman, R. Dynamic programming. 1954. 33

Blum, A. and Mitchell, T. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998. 72

Boersma, P. Praat, a system for doing phonetics by computer. *Glot international*, 5(9/10):341–345, 2002. 81

Bohus, D. and Horvitz, E. Models for multiparty engagement in open-world dialog. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 225–234. Association for Computational Linguistics, 2009. 78

Bohus, D. and Rudnicky, A. Constructing accurate beliefs in spoken dialog systems. In *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, pages 272–277. IEEE, 2005. 29

Boite, R., Bourlard, H., Dutoit, T., Hancq, J., and Leich, H. *Traitement de la Parole*, page 488. Number 2-88074-388-5. Presses Polytechniques Universitaires Romandes Lausanne, 2nd edition edition, 2000. 12

Bos, J., Klein, E., Lemon, O., and Oka, T. Dipper: Description and formalisation of an information-state update dialogue system architecture. In *4th SIGdial Workshop on Discourse and Dialogue*, pages 115–124, 2003. 28

Bourlard, H. and Morgan, N. *Connectionist speech recognition: a hybrid approach*. Springer, 1994. 11

Boyen, X. and Koller, D. Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 33–42. Morgan Kaufmann Publishers Inc., 1998. 46

Bradtke, S. and Barto, A. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(1):33–57, 1996. 51

Brown, P., Desouza, P., Mercer, R., Pietra, V., and Lai, J. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992. 60

Bui, T., Poel, M., Nijholt, A., and Zwiers, J. A tractable hybrid ddn–pomdp approach to affective dialogue modeling for probabilistic frame-based dialogue systems. *Natural Language Engineering*, 15(02):273–307, 2009. 48

Chomsky, N. *Aspects of the Theory of Syntax*, volume 119. MIT Press (MA), 1965. 14

Davis, K., Biddulph, R., and Balashek, S. Automatic recognition of spoken digits. *Journal of Acoustic Society of America*, 24(6):637–642, 1952. 9

Dumouchel, P., Gupta, V., Lennig, M., and Mermelstein, P. Three probabilistic language models for a large-vocabulary speech recognizer. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pages 513–516. IEEE, 1988. 60

Evermann, G. and Woodland, P. Large vocabulary decoding and confidence estimation using word posterior probabilities. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, volume 3, pages 1655–1658. IEEE, 2000. 22

Fodor, P. and Huerta, J. Planning and logic programming for dialog management. In *Spoken Language Technology Workshop, 2006. IEEE*, pages 214–217. IEEE, 2006. 28

Forbes-Riley, K. and Litman, D. Adapting to student uncertainty improves tutoring dialogues. In *Proc. Intl. Conf. on Artificial Intelligence in Education*, 2009. 79

Fosler-Lussier, E. and Kuo, H. Using semantic class information for rapid development of language models within asr dialogue systems. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 1, pages 553–556. IEEE, 2001. 60

Ghani, R. Combining labeled and unlabeled data for multiclass text categorization. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 187–194, 2002. 70

Goddeau, D., Meng, H., Polifroni, J., Seneff, S., and Busayapongchai, S. A form-based dialogue manager for spoken language applications. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 2, pages 701–704. IEEE, 1996. 27

Grosz, B. The structure of task oriented dialogs. In *IEEE Symposium on Speech Recognition: Contributed Papers. Carnegie Mellon University Computer Science Dept., Pittsburgh, Pennsylvania*, 1974. 10

Grosz, B. and Sidner, C. Attention, intentions, and the structure of discourse. *Computational linguistics*, 12(3):175–204, 1986. 10

Gunawardana, A., Mahajan, M., Acero, A., and Platt, J. Hidden conditional random fields for phone classification. In *Ninth European Conference on Speech Communication and Technology*, 2005. 79

He, X., Deng, L., and Chou, W. Discriminative learning in sequential pattern recognition. *Signal Processing Magazine, IEEE*, 25(5):14–36, 2008. 79

He, Y. and Young, S. Spoken language understanding using the hidden vector state model. *Speech Communication*, 48(3):262–275, 2006. 23

Heather, P., Stuart M, S., et al. Recognizing uncertainty in speech. *EURASIP Journal on Advances in Signal Processing*, 2011, 2011. 79

HENDERSON, J. AND LEMON, O. Mixture model pomdps for efficient handling of uncertainty in dialogue management. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 73–76. Association for Computational Linguistics, 2008. 49

HIFNY, Y. AND RENALS, S. Speech recognition using augmented conditional random fields. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17(2):354–365, 2009. 79

HINTON, G., OSINDERO, S., AND TEH, Y. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. 79

JELINEK, F. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64(4): 532–556, 1976. 10

JELINEK, F. *Statistical methods for speech recognition*. the MIT Press, 1997. 58

JELINEK, F., MERIALDO, B., ROUKOS, S., AND STRAUSS, M. A dynamic language model for speech recognition. In *Proceedings of the DARPA Workshop on Speech and Natural Language*, pages 293–295, 1991. 60

JIANG, H. Confidence measures for speech recognition: A survey. *Speech communication*, 45(4): 455–470, 2005. 22

JIANG, H. AND LI, X. Incorporating training errors for large margin hmms under semi-definite programming framework. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–629. IEEE, 2007. 79

JUANG, B., HOU, W., AND LEE, C. Minimum classification error rate methods for speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 5(3):257–265, 1997. 79

JUNG, S., LEE, C., KIM, K., JEONG, M., AND LEE, G. Data-driven user simulation for automated evaluation of spoken dialog systems. *Computer Speech & Language*, 23(4):479–509, 2009. 26

JURAFSKY, D. AND MARTIN, J. Speech and language processing: An introduction to speech recognition. *Computational Linguistics and Natural Language Processing. 2nd Edn., Prentice Hall, ISBN*, 10(0131873210):794–800, 2008. 15

JURCICEK, F., GAŠIC, M., KEIZER, S., MAIRESSE, F., THOMSON, B., YU, K., AND YOUNG, S. Transformation-based learning for semantic parsing. 2009. 23

KAPADIA, S., VALTCHEV, V., AND YOUNG, S. Mmi training for continuous phoneme recognition on the timit database. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, volume 2, pages 491–494. IEEE, 1993. 79

KEARNS, M., LITMAN, D., SINGH, S., AND WALKER, M. Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *Arxiv preprint arXiv:1106.0676*, 2011. 31

Krahmer, E. and Swerts, M. How children and adults produce and perceive uncertainty in audiovisual speech. *Language and speech*, 48(1):29–53, 2005. 79

Kschischang, F., Frey, B., and Loeliger, H. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47(2):498–519, 2001. 42, 44

Kuhn, R. Speech recognition and the frequency of recently used words: A modified markov model for natural language. In *Proceedings of the 12th conference on Computational linguistics-Volume 1*, pages 348–350. Association for Computational Linguistics, 1988. 61

Kupiec, J. Probabilistic models of short and long distance word dependencies in running text. In *Proceedings of the workshop on Speech and Natural Language*, pages 290–295. Association for Computational Linguistics, 1989. 61

Lee, C. and Narayanan, S. Toward detecting emotions in spoken dialogs. *Speech and Audio Processing, IEEE Transactions on*, 13(2):293–303, 2005. 80

Lemon, O., Georgila, K., Henderson, J., and Stuttle, M. An isu dialogue system exhibiting reinforcement learning of dialogue policies: generic slot-filling in the talk in-car system. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Posters & Demonstrations*, pages 119–122. Association for Computational Linguistics, 2006. 31

Levin, E. and Pieraccini, R. A stochastic model of computer-human interaction for learning dialogue strategies. In *Fifth European Conference on Speech Communication and Technology*, 1997. 30

Levinson, S., Rabiner, L., and Sondhi, M. An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. *Bell Syst. Tech. J*, 62 (4):1035–1074, 1983. 61

Liscombe, J., Hirschberg, J., and Venditti, J. Detecting certainness in spoken tutorial dialogues. In *Ninth European Conference on Speech Communication and Technology*, 2005. 81

Litman, D. and Forbes, K. Recognizing emotions from student speech in tutoring dialogues. In *Automatic Speech Recognition and Understanding, 2003. ASRU'03. 2003 IEEE Workshop on*, pages 25–30. IEEE, 2003. 13

Litman, D. and Silliman, S. Itspoke: An intelligent tutoring spoken dialogue system. In *Demonstration Papers at HLT-NAACL 2004*, pages 5–8. Association for Computational Linguistics, 2004. 22

Mairesse, F. and Walker, M. Personage: Personality generation for dialogue. In *Annual Meeting-Association For Computational Linguistics*, volume 45, page 496, 2007. 25

MAIRESSE, F., GASIC, M., JURCICEK, F., KEIZER, S., THOMSON, B., YU, K., AND YOUNG, S. Spoken language understanding from unaligned data using discriminative classification models. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 4749–4752. IEEE, 2009. 23

MCDERMOTT, E., HAZEN, T., LE ROUX, J., NAKAMURA, A., AND KATAGIRI, S. Discriminative training for large-vocabulary speech recognition using minimum classification error. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(1):203–223, 2007. 79

MCTEAR, M. ?spoken dialogue technology: Toward the conversational user interface.? *University of Ulster) London Springer-Verlag*, 2004. 15

MINKA, T. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001. 46

MURPHY, K. *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, University of California, 2002. 46

NIGAM, K., MCCALLUM, A., THRUN, S., AND MITCHELL, T. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2):103–134, 2000. 67, 70

PAEK, T. AND CHICKERING, D. Evaluating the markov assumption in markov decision processes for spoken dialogue management. *Language resources and evaluation*, 40(1):47–66, 2006. 31

PAEK, T. AND JU, Y. Accommodating explicit user expressions of uncertainty in voice search or something like that. In *Ninth Annual Conference of the International Speech Communication Association*, 2008. 79

PETERS, J. AND SCHAAL, S. Natural actor-critic. *Neurocomputing*, 71(7):1180–1190, 2008. 51, 52

PIERACCINI, R. AND HUERTA, J. Where do we go from here? research and commercial spoken dialog systems. In *6th SIGdial Workshop on Discourse and Dialogue*, 2005. 27

POVEY, D. Discriminative training for large vocabulary speech recognition. *Cambridge, UK: Cambridge University*, 2004. 79

POVEY, D. AND WOODLAND, P. Minimum phone error and i-smoothing for improved discriminative training. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 1, pages I–105. IEEE, 2002. 79

POVEY, D., KANEVSKY, D., KINGSBURY, B., RAMABHADRAN, B., SAON, G., AND VISWESWARIAH, K. Boosted mmi for model and feature-space discriminative training. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 4057–4060. IEEE, 2008. 79

PUTERMAN, M. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, Inc., 1994. 29

RABINER, L. AND SCHAFER, R. *Digital processing of speech signals*, volume 100. Prentice-hall Englewood Cliffs, NJ, 1978. 10

RANGANATH, R., JURAFSKY, D., AND MCFARLAND, D. It's not you, it's me: detecting flirting and its misperception in speed-dates. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 334–342. Association for Computational Linguistics, 2009. 80

RAUX, A., BOHUS, D., LANGNER, B., BLACK, A., AND ESKENAZI, M. Doing research on a deployed spoken dialogue system: one year of let's go! experience. In *Ninth International Conference on Spoken Language Processing*, 2006. 21

RICH, C. AND SIDNER, C. Collagen: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, 8(3):315–350, 1998. 28

ROSENBERG, C., HEBERT, M., AND SCHNEIDERMAN, H. Semi-supervised self-training of object detection models. 2005. 70

ROY, N., PINEAU, J., AND THRUN, S. Spoken dialogue management using probabilistic reasoning. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 93–100. Association for Computational Linguistics, 2000. 31

SCHATZMANN, J., WEILHAMMER, K., STUTTLE, M., AND YOUNG, S. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review*, 21(2):97–126, 2006. 26

SCHATZMANN, J., THOMSON, B., AND YOUNG, S. Error simulation for training statistical dialogue systems. In *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pages 526–531. IEEE, 2007. 26

SEARLE, J. *Speech acts: An essay in the philosophy of language*. Cambridge Univ Pr, 1969. 22

SEYMORE, K., MCCALLUM, A., AND ROSENFELD, R. Learning hidden markov model structure for information extraction. In *AAAI-99 Workshop on Machine Learning for Information Extraction*, pages 37–42, 1999. 66

SHA, F. AND SAUL, L. Large margin gaussian mixture modeling for phonetic classification and recognition. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 1, pages I–I. IEEE, 2006. 79

SOLSONA, R., FOSLER-LUSSIER, E., KUO, H., POTAMIANOS, A., AND ZITOUNI, I. Adaptive language models for spoken dialogue systems. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, volume 1, pages I–I. IEEE, 1993. 59

SUTTON, R. AND BARTO, A. *Reinforcement learning: An introduction*, volume 1. Cambridge Univ Press, 1998. 51

THEDE, S. AND HARPER, M. A second-order hidden markov model for part-of-speech tagging. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 175–182. Association for Computational Linguistics, 1999. 62

TRAUM, D. Speech acts for dialogue agents. *Foundations of rational agency*, 14:169–201, 1999. 22

TUR, G., HAKKANI-TÜR, D., AND SCHAPIRE, R. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45(2):171–186, 2005. 66

VITERBI, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269, 1967. 62

WALKER, M. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *Arxiv preprint arXiv:1106.0241*, 2011. 31

WALKER, W., LAMERE, P., KWOK, P., RAJ, B., SINGH, R., GOUVEA, E., WOLF, P., AND WOELFEL, J. Sphinx-4: A flexible open source framework for speech recognition. 2004. 21

WILLIAMS, J. Using particle filters to track dialogue state. In *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pages 502–507. IEEE, 2007. 49

WILLIAMS, J. AND YOUNG, S. Scaling up pomdps for dialog management: The“summary pomdp”method. In *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, pages 177–182. IEEE, 2005. 39, 50

WONG, Y. AND MOONEY, R. Learning synchronous grammars for semantic parsing with lambda calculus. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 960, 2007. 23

YAROWSKY, D. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics, 1995. 70

YOUNG, S., KERSHAW, D., ODELL, J., OLLASON, D., VALTCHEV, V., AND WOODLAND, P. The {HTK} book version 3.0. 2000. 21

YOUNG, S., GASIC, M., KEIZER, S., MAIRESSE, F., SCHATZMANN, J., THOMSON, B., AND YU, K. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174, 2010. 48, 49

YU, D. AND DENG, L. Deep-structured hidden conditional random fields for phonetic recognition. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010. 79

Yu, D., Deng, L., He, X., and Acero, A. Use of incrementally regulated discriminative margins in mce training for speech recognition. In *Ninth International Conference on Spoken Language Processing*, 2006. 79

Zettlemoyer, L. and Collins, M. Online learning of relaxed ccg grammars for parsing to logical form. In *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL-2007*. Citeseer, 2007. 23

Zhou, D., He, Y., and Kwoh, C. Extracting protein-protein interactions from the literature using the hidden vector state model. *Computational Science–ICCS 2006*, pages 718–725, 2006. 66

Zhou, D., He, Y., and Kwoh, C. Semi-supervised learning of the hidden vector state model for extracting protein-protein interactions. *Artificial Intelligence in Medicine*, 41(3):209–222, 2007. 66

Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T., and Hetherington, L. Juplter: a telephone-based conversational interface for weather information. *Speech and Audio Processing, IEEE Transactions on*, 8(1):85–96, 2000. 15

Zweig, G. and Nguyen, P. Scarf: A segmental conditional random field toolkit for speech recognition. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010. 79

PUBLICATIONS

Refereed Journals / Conference Papers /Presentation/Invited talks

2012 **Efficient Dialogue Management Of A Spoken Dialogue System By Using Prosody Information Of The User Utterances** (Manzoor Ahmad Chachoo and Dr. S M K Quadri), *International Journal of Engineering Research and Applications*, Volume 2, Number 4, Pages 1700-1703, eISSN: 2248-9622, July-August 2012.

2012 **Semi-Supervised learning of utterances using hidden vector state language model** (Manzoor Ahmad Chachoo, Dr. S.M.K. Quadri), *Journal of Global Research in Computer Science*, Volume 3, Issue 6, Pages 70-74, pISSN: 2229-371X, June 2012.

2012 **Adaptive Hybrid POS Cache based Semantic Language Model** (Manzoor Ahmad Chachoo and Dr. S M K Quadri), *International Journal of Computer Applications*, Volume 39, Number 13, Pages 07-10, eISSN: 0975 8887, February 2012.

2011 **Morphological analysis of Kashmiri language using open source Extract tool** (Manzoor Ahmad Chachoo, Dr.S.M.K. Quadri), *Trends in Information Management*, Volume 7, Issue 2, Pages 176-187, eISSN: 0973-4163, December 2011.

2011 **Morphological analysis of Kashmiri language using open source Extract tool**, *Presented at National Seminar on Open Source Softwares: Challenges & Opportunities*, June 20-22, University of Kashmir, India, 2011.

2011 **Some issues in the development of spoken language interface plugin for open source operating systems**, *Presented at National Seminar on Open Source Softwares: Challenges & Opportunities*, June 20-22, University of Kashmir, India, 2011.

2008 **Datawarehouse for Salary Management System** , *Presented at* 6th *JK Science Congress*, December, University of Kashmir, India, 2008.

2007 **Ambiguity Resolution in Machine Translation: Some Issues in Kashmiri to English Translation**, *Presented at EIGHTH INTERNATIONAL CONFERENCE ON SOUTH ASIAN LANGUAGES (ICOSAL-08)*, Dept. of Linguistics, Aligarh Muslim University, Aligarh in collaboration with Central Institute of Indian Languages, (CIIL), Mysore, January 06-08, 2008.