5-1-2010

Artificial Conversations for Chatter Bots Using Knowledge Representation, Learning, and Pragmatics

Chayan Chakrabarti

Chayan Chakrabarti

Candidate

Computer Science

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

George Luger , Chairperson

Thomas Caudell

Lydia Tapia

Chuck Wooters

Jessica Turner

Artificial Conversations for Chatter Bots Using Knowledge Representation, Learning, and Pragmatics

by

Chayan Chakrabarti

B.E., Computer Engineering, University of Mumbai, 2001
M.S., Computer Science, University of New Mexico, 2005

DISSERTATION

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy
Computer Science

The University of New Mexico

Albuquerque, New Mexico

May, 2014

Dedication

To my parents.

"A computer would deserve to be called intelligent if it could deceive a human into believing that it was human" – Alan Turing

Acknowledgments

My highest gratitude is for my advisor, Professor George Luger, for introducing me to the world of Artificial Intelligence research and giving me an opportunity to be part of it. He encouraged me to return to graduate school, taught me to have a vision and philosophy, kept me focussed on the big picture, guided me through every step of the dissertation process, and enriched my academic experience.

My committee members patiently helped me through all the phases of the dissertation and gave me enthusiastic feedback at very short notice. I thank Professor Thomas Caudell for always being a great mentor, Professor Lydia Tapia for her invaluable insights, Dr. Chuck Wooters for helping me germinate the ideas for this dissertation, and Professor Jessica Turner for her constant support.

Several people helped me throughout the program. I thank Professor Caroline Smith and Professor Joel Castellanos for their help with my dissertation. I thank Professor Frank Gilfeather for supporting my early research. I have interacted with several outstanding faculty and gained important skills under their tutelage. I will forever be grateful to them. I especially thank Professor Deepak Kapur and Professor David Ackley.

I feel I learned as much from my peers as I did from my classes. I sincerely thank my lab mates at the AI Lab, Benjamin Gordon, Sunny Fugate, Thomas Jones, and all the others for always being so supportive. I thank all my other colleagues and friends for making me a better student, researcher, and person. I thank everyone at the System Support Group and the Front Office for making my life so much easier.

The graduate school journey is an emotional roller coaster. I feel that my parents experienced the highs and lows even more intensely than I did. I am so thankful to my parents for always being there for me.

Artificial Conversations for Chatter Bots Using Knowledge Representation, Learning, and Pragmatics

by

Chayan Chakrabarti

B.E., Computer Engineering, University of Mumbai, 2001
M.S., Computer Science, University of New Mexico, 2005

Ph.D., Computer Science, University of New Mexico, 2014

Abstract

Chatter bots are software programs that engage in artificial conversations through a text-based input medium. Many businesses have automated their online customer service support by deploying chatter bots. These customer service chatter bots interact with customers, answer their queries, and address service related issues.

Traditional chatter bots perform best in artificial conversations consisting of pairs of utterance exchanges such as question-answer sessions, where the context may or may not switch with every exchange pair. They perform poorly in longer conversations, where the context is maintained over several pairs of utterance exchanges. Existing approaches to artificial conversation generation focus on linguistic and grammatical modeling using natural language processing and computational linguistics techniques to generate individual sentence-level utterances.

This research explores techniques to go beyond individual sentence-level interactions to model the higher level conversation process. A conversation is a process that adheres to well-defined semantic conventions and is contextually grounded in domain-specific knowledge. This dissertation presents a modular, robust, and scalable architecture that combines content semantics and pragmatic semantics to generate higher quality artificial conversations in the customer service domain. The conversational process is modeled using stochastic finite state machines, where the parameters of the model are learned from a corpus of human conversations.

This research leverages specific concepts from conversation theory and speech act theory. For evaluation purposes, the artificial conversations are graded by a panel of human judges using criteria that include Grice's cooperative maxims.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Chatter Bots and Artificial Conversations

Chatting with computer programs has come a long way, from pioneering artificial intelligence demonstrations like ELIZA to modern intelligent personal assistant software like Siri. Early chatter bots were designed as academic testing tools for natural language processing theories. Lately, chatter bots have become more sophisticated. They have found applications in interactive games, as website navigation tools, and for simulating personal help desk assistants. It is estimated that by 2015, at least 50% of customer service will be realized via chatter bots, and this can result in cost reductions up to 20% along with increased customer loyalty [24]. But have chatter bots scaled the zenith of their ontogenesis, or are there opportunities to extend their current capabilities? This thesis identifies a key limitation of current contemporary chatter bots, and presents a solution to address it.

1.1 Internet Chatting and Chatter Bots

To understand chatter bots, it is important to understand online chatting as a medium of communication. Why is this medium so popular? It has gained popularity because it is non-invasive, unlike a phone call, and at the same time it is inherently an instant medium of exchange. Communicators are not compelled to respond instantly, but can carry on a conversation in near instant time if they desire. The chat logs can be stored and easily referenced at a future point in time. The medium is easily available to anyone with access to an internet enabled computer or mobile device. The inherently informal nature of the medium lowers inhibitions, encourages spontaneity, and generally reduces the barriers to effective communication.

For these very reasons, enterprise applications started leveraging the medium of online chatting. One of the earliest enterprise applications of chatting was in IT support [34]. System users could chat with their system administrators and get help with their IT related issues. Network administrators could also chat with each other and exchange information and advice. Computer programmers also extensively used chat rooms to exchange information about bugs, give advice to newbie programmers, and discuss pertinent programming issues. The chat logs were archived on servers and could be easily retrieved in future for reference. The chat transcript logs contained a wealth of information and served as a virtual FAQ for many IT and programming related queries [34]. Many businesses capitalized on the growth of chat as a communication medium and implemented their customer service support operations through chat interfaces. Existing and potential customers could chat with customer service representatives and resolve their customer service issues or seek information about the business' products and services [24].

It was not long before the artificial intelligence community started creating bots to emulate human chatting. Classic chatterbots like ELIZA [98] and PARRY [12] were designed with academic intent, to understand and test theories in natural language processing. Later efforts in chatter bot evolution focussed on making chatter bots more human-like. The Loebner Prize contest began in 1990, rewarding chatter bots considered to be the most human-like [48]. Many famous chatter bots like A.L.I.C.E. [97], Albert One [23], George [4], Rosette [100], Chip Vivant [18], and Mitsuku [103] were born out this effort [77]. These endeavors contributed to pushing the boundaries of computational linguistics and natural language processing.

Of late, chatter bots have evolved to a sufficient level of maturity, and consequently they have been leveraged in several real-world applications. Many businesses and organizations have now implemented chatter bots to simulate human customer service representatives. Customers can now chat directly with the chatter bot, without any human intervention. These chatter bots often have virtual personalities and can closely imitate a human representative during a conversation.

For example, when customers go to the Alaska Airlines homepage, they have an option to "Ask Jenn" (`www.alaskaair.com`). Jenn is a chatter bot who helps customers purchase flight tickets on Alaska Airlines. She can answer a range of questions regarding Alaska Airlines like making reservations, payment options, changing reservations, and general inquiries. Similarly, the US Army recruitment homepage (`www.goarmy.com`) has a chatter bot named Sgt. Star. He can answer common queries about joining the Army. In addition to answering questions, the chatter bots can also open up relevant websites that might have the information that the customer or inquirer is seeking. (See Figure 1.1)

Similarly, an electronics company can have a chatter bot helping a customer troubleshoot problems with a gadget, or direct him to other resources to solve the problem. An insurance company can help a potential customer select an insurance plan customized to her unique needs. Businesses realize an economic benefit from customer service chatter bots. Compared to human representatives, chatter bots can

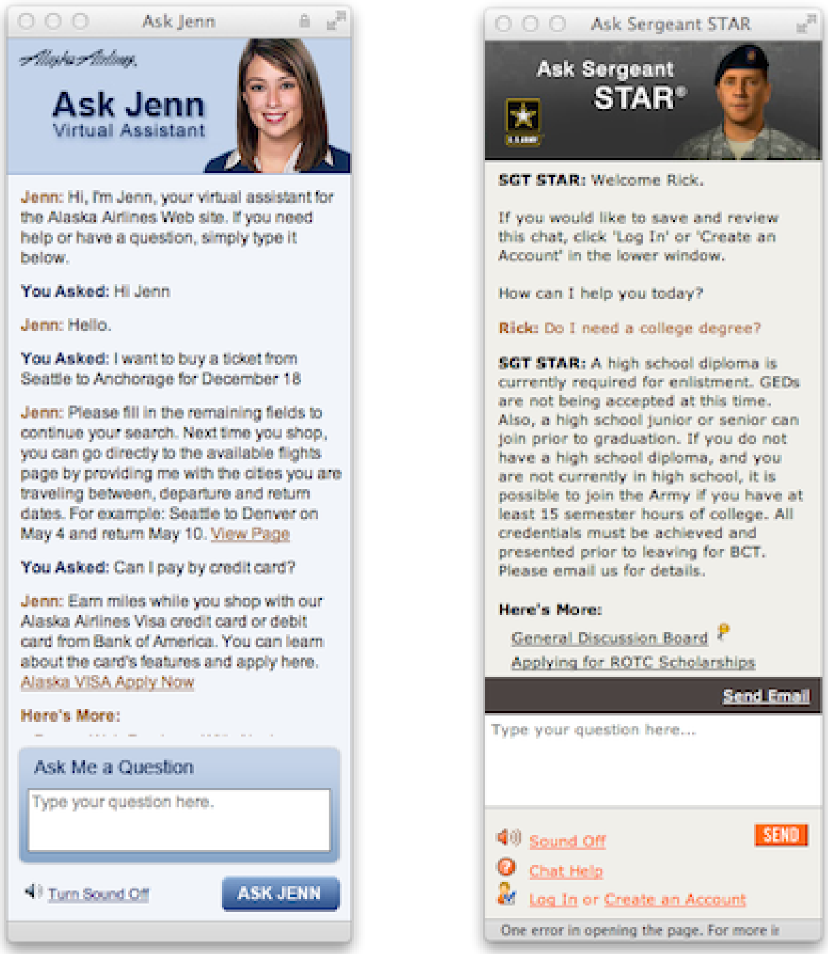

Figure 1.1: Virtual Chatter Bots Jenn and Sgt. Star, deployed by Alaska Airlines and United States Army respectively. Customers can engage these chatter bots in a conversation, and seek their responses to specific questions.

Source: Alaska Airlines & Go Army

be more reliable, make fewer mistakes, can be easily scaled to work larger loads, can be modified without extensive retraining, and in many instances, they can work in parallel.

The story of chatter bot evolution is now at an interesting juncture. Contemporary chatter bots are sophisticated enough to leverage cutting age concepts like semantic parsing [58] and sentiment analysis [99]. What could be the next step in chatter bot evolution? Examining limitations of current chatter bots is the first step in trying to answer this question. In the rest of this chapter, I will highlight with examples some specific deficiencies of chatter bot technology. I will then formalize what it means to design an appropriate solution.

1.2 Limitations of Chatter Bots

Using some examples, I differentiate the concept of chatting as in merely uttering something, and holding a conversation, which implies something more involved. This section develops this idea with some illustrations.

Consider the chatter bot Mitsuku [103] (`http:/http://mitsuku.com`), who was the winner of the 2013 Loebner Prize. Observe some snippets of actual conversation with Mitsuku.

Mitsuku : `Good evening. My name is Mitsuku. What is your name?`

Human : `My name is Sally.`

Mitsuku : `Nice to meet you, Sally.`

Human : `Can you recommend a good store for designer clothing?`

Mitsuku : `I mostly wear jeans and a t-shirt.`

Human : `But I need a dress for a formal reception I am attending.`

Mitsuku : `Do you tend to like most things?`

Human : `No.`

Mitsuku : `OK. So, what do you do for a living?`

Human : `I am an accountant.`

Mitsuku : `What do you predict for the economy this year - bad or really bad?`

While Mitsuku seems articulate enough, the conversation clearly did not go the way Sally, the human, probably expected. So what exactly went wrong? Consider another actual conversation snippet with Mitsuku [103].

Mitsuku : `Welcome back. How are you today?`

Human : `Not too good. I have to make a tough decision.`

Mitsuku : `That's too bad.`

Human : `Can you help me?`

Mitsuku : `Not often. Can you?`

Human : `I need your advise with a personal dilemma.`

Mitsuku : `Do you find yourself envious of what others own?`

Human : `No.`

Mitsuku : `Why not? Do you live around here?`

Human : `Yes.`

Mitsuku : `Do you have any pets?`

Human : `No.`

Mitsuku : `That's a definite no? I travel too much to really have pets. We do have two chickens in the backyard.`

Clearly Mitsuku did not do too well with this conversation. Although the state of the art in chat bot technology, the conversation still seems unsatisfactory. How do we quantify what is going wrong? If we observe the conversation closely, we notice a definite pattern. When Mitsuku's response is evaluated only in relation to the immediate previous utterance by the human, she grades satisfactorily. It is only when evaluated on a longer sustained conversation, that she grades poorly. Mitsuku performs adequately in an isolated question-answer exchange. She does well even over a series of several consecutive question-answer pairs. However, a series of question-answer pairs, or a series of one-to-one utterances, does not constitute a conversation. Holding a conversation would be the next step in the evolution of chatter bots, but the current state of the art chatter bot is unable to perform this task.

1.3 Customer Service Chatter Bots

Chatter bots like Mitsuku [103] are general purpose chatter bots, i.e., they attempt to hold open conversations in no specific area. Clearly this seems like a very difficult task computationally. But what if the scope was restricted? Would chatter bots be able to perform better?

Section 1.1 mentioned customer service chatter bots, with some examples. Indeed, these chatter bots exist in many forms, from website navigation assistants, to software virtual helpers, to gaming agents deployed extensively in massively multiplayer online role playing gaming engines, to more recognizable ones like Siri on Apple's mobile platform. All of these are not strictly chatter bots by the convention we are following here. Siri doesn't merely chat, but speaks. But the basic task being performed is still a conversation. Hence for illustration purposes if would be appropriate to evaluate them and compare them to chatter bots like Mitsuku. These bots typically would operate, i.e., hold conversations in a domain restricted to the company that was deploying them. Thus, their task should appear to be less computationally challenging than Mitsuku. So how well do they fare?

Let me illustrate another example from a real-world chatter bot application to highlight this deficiency. Consider a snippet of conversation with a customer service chatter bot that helps customers solve account access issues. In this exchange, the customer is unable to login to an account, and seeks assistance.

Chatter Bot : `Hello. How can I help you today?`

Customer : `I am unable to login to my account. I forgot my password.`

Chatter Bot : `Please click on the "Forgot Password" link. You will be emailed a link that will help you change your password.`

Customer : `I already tried that. That link takes me to a form to reset password. But it requires me to enter my Customer Relationship Number. I do not have this number.`

Chatter Bot : `The Customer Relationship Number can be found in the "Account Information" tab when you log in to your account.`

Customer : `But I am unable to login to my account. I do not remember my password.`

Chatter Bot : `Please click on the "Forgot Password" link. You will be emailed a link that will help you change your password.`

Once again, one can observe that while the chatter bot can carry on a question-answer session satisfactorily, it is unable to determine if the answer was satisfactory for the customer. It is unable to tell if its response adequately addressed the customer's query, or if the customer was confused by the response.

To a human representative, it would be clear that this conversation is not helping address the customer's issue. However, since chatter bots are programmed through pattern-template pairs, they are more suited for question-answer type exchanges, where there is a sequence of several question-answer pairs. They are unable to hold a longer conversation, understand the conversation, gauge whether the conversation is going in the desired direction, and act on it. In the above example, we expect the chatter bot to recognize the deadlock situation for the customer and to resolve it, or transfer the conversation to a human representative if the solution is beyond its programmatic capabilities.

Let me use another example to illustrate the subtle failure of such customer service chatter bots. In this example, a chatter bot deployed in customer service operations of an online electronic trading website tries to help a customer resolve an unallocated transaction, i.e., the customer's margin account does not have enough balance to service a pending buy order. (Acknowledgement: This corpus was provided by Next IT Inc., Spokane, WA.)

Chatter Bot: `Hello. How can I help you today?`

Customer: `I have a pending buy order and I am out of margin.`

Chatter Bot: `Buy orders are processed within 2 business days.`

Customer: `But I am out of margin. I am concerned that my buy order will not execute.`

ChatterBot: `Margin can be restored by a new allocation request from your holding account. It takes 3 business days to execute.`

Customer: `But my buy order is already scheduled. I do not want to lose it.`

Chatter Bot: `Buy orders are processed within 2 business days.`

It is clear from these examples that the conversational performance of these chatter bots does not live up to what we would expect from a human customer service agent. Hence, there is clearly an opportunity to improve their conversational ability.

In addition, since chatter bots cannot actually follow a longer conversation, they are also incapable of spotting opportunities in a conversation for providing unsolicited but pertinent information. Consider the following exchange between a prospective insurance customer and a chatter bot.(Acknowledgement: This corpus was provided by Next IT Inc., Spokane, WA.)

Chatter Bot : `Hello. How can I help you today?`

Customer : `I received an quote for home insurance last week. I would like to purchase that policy.`

Chatter Bot : `All right. Do you remember the quote reference number?`

Customer : `No. I had printed it out but I left the printout in my car.`

Chatter Bot : `Not a problem. I can look it up by your last name and date of birth. What is your last name?`

Customer : `Johnson`

Chatter Bot : `What is your date of birth?`

Customer : `April 21, 1973`

Chatter Bot : `Great. I found your quote. Would you like to purchase the policy now?`

It would seem that the chatter bot satisfactorily handled the conversation. However, there are some missed opportunities. The conversation indicates that the customer possibly owns a car. A human representative would have noticed this, and spotted an opportunity for another sale.

Human Representative : `You mentioned that you have a car. Did you know that if you get auto insurance in conjunction with your home insurance, it could save you $500 over 6 months?`

The next generation in the evolution of chatter bots should be able spot such opportunities in a conversation, and act on it, either by disseminating relevant information, or by transferring the conversation to a human representative seamlessly. The missing element in current chatter bots seems to be one of context. In a series of question-answer exchanges, or pairwise utterance, the context switches from one pair to the next. But in most conversations, the context remains the same throughout the exchange of utterances. Contemporary chatter bots are unable to adhere to context in conversations. It is this limitation that I will explore further and address in this research.

Conversations abound with uncertainties of various kinds that may lead to misunderstanding and other communication failures. These uncertainties permeate every level of conversation, from attending to what was said and identifying what words were spoken, to understanding the intentions behind the words. While human representatives manage these multiple uncertainties with almost effortless ease, chatter bots often break down in these situations. My aim is to improve the conversational power of chatter bots. Instead of just being able to engage in question-answer exchanges, I would like them to be able to hold a longer conversation, be able to semantically process it, and more closely to emulate the behavior of a human representative.

In the rest of the chapter, I will highlight many relevant nuances to this problem and formalize the definition of this problem. What exactly does it mean for the chatter bot to better understand context? How exactly can context be represented formally? What would be the evaluation parameters for this research endeavor? What underlying assumptions about scope should be made to control the evaluation parameters in this research? What other concepts from existing literature in natural language processing, computational linguistics, and semantic learning can be leveraged for this research? The rest of the chapter will be dedicated to answering these questions.

What are the types of conversation that chatter bots can carry out? Many chatter bots have been deployed in real-life that can carry out conversations ranging from open ended general chitchat to specific questions-answer sessions. In order

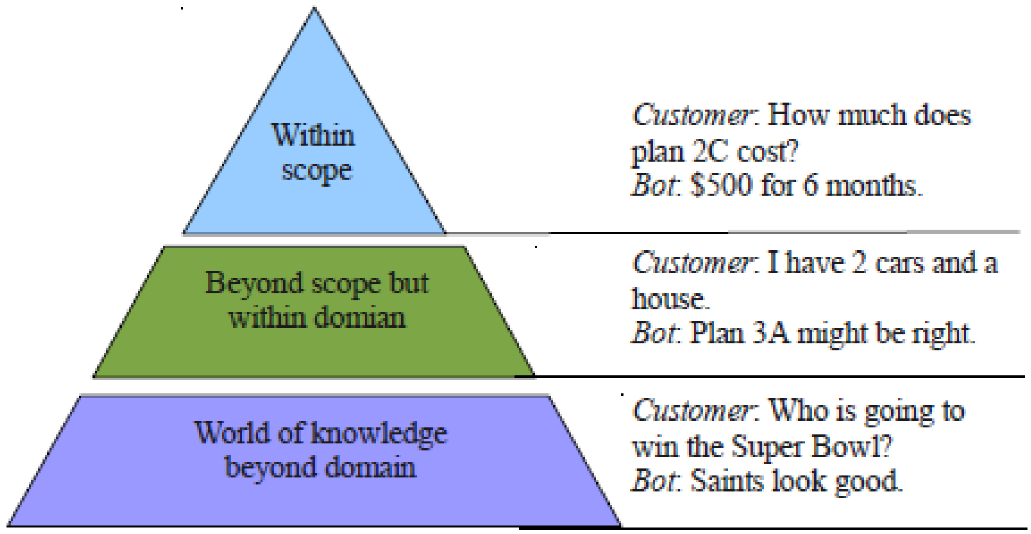

Figure 1.2: Classes of Conversations: *Within scope* denoting specific conversations within the domain, *Beyond scope* denoting non-specific open-ended conversations within the domain, and *Beyond domain* denoting conversations beyond the domain being considered.

Acknowledgement: Charles Wooters. Next IT Corporation (2011).

to achieve measurable success in my goal of creating better chatter bots, I have to restrict the domain of conversations to consider. Hence, I am restricting my conversations to customer service contexts, in which a customer calls up a business, engages the chatter bot, and seeks resolution for customer service problems. Even in this domain, the range of conversations is too large for full analysis. My observation of real-life customer service chat logs leads me to assert that there is a pyramid-type distribution to the conversation space, as depicted in Figure 1.2. The classes of conversations in the bottom tier of the pyramid, can include anything in the whole space of conversations. This includes the entire world of knowledge, not restricted to a particular domain. The middle tier of the pyramid, includes conversations that fall within a specific domain, but can be open ended in nature. The top tier of the pyramid includes conversations that are restricted to the specific domain, and are very precise in nature. In my analysis, I shall restrict the space of conversations to the top two tiers of this pyramid.

Currently, when organizations deploy chatter bots for customer service, the chatter bots are programmed with knowledge that is essential to perform its task of customer service. This process involves encoding rules about the organization's policies, products, and services. This can be generalized as a knowledge representation problem. In the field of information organization and extraction, there are several proven data structures for the effective knowledge representation. They range from simple graphs to an ontology, where there is a formal representation of knowledge as a set of concepts within a domain and the relationships between those concepts. It is used to reason about the entities within that domain, and may be used to describe the domain. An ontology provides a shared vocabulary, which can be used to model a domain, the type of objects and/or concepts that exist, and their properties and relations. Information retrieval using ontology engineering techniques automatically extracts structured and categorized information that is contextually and semantically well-defined from a specific domain from unstructured machine-readable documents. These techniques are useful for richer extraction of information from sources that are a combination of structured and unstructured data [21].

How exactly can a conversation be represented in formal terms? A conversation can have specific states and progressively transition through these states. The states can be indicators of the quality of the conversation, for example, is the customer satisfied or dissatisfied, is the customer getting the relevant information, or getting confused, is the conversation at a suitable point where it is advisable to reveal new information to the customer, etc. [26, 89]. The sentiment analysis community tries to gauge parameters like political opinion, favorability or unfavorability of products or services, etc. based on twitter chatter or blog posts. However gauging sentiment in a conversation is far more subtle. Generally customers do not explicitly use words that signify positive or negative sentiment. The deduction that the customer is satisfied or unsatisfied can only be made by measuring the direction of flow of a conversation through various states. In this research, I will define a conversation as a state-based process. The conversation control algorithm will predict the flow of the conversation through a transition probability matrix, and the chatter bot's responses will be engineered to steer the conversation to more favorable states [64, 65].

It is important to define a validation criterion for this research. It will help determine if I have made progress and succeeded in the goals of this research. The validation criterion will depend on a formal scheme to measure the suitability of a conversation. I will define one such measurement scheme, Grice's maxim's of conversations [30, 31, 32, 51], later in this document. These will be used to validate the results of my research.

1.4 Artificial Conversations

The main aim of this research to explore and demonstrate techniques for solving the problem of artificial conversations. This work defines artificial conversations as any kind of simulated interactive conversation between a software entity and a human user. While this research will restrict itself to text based chat conversations, in which all interaction between the software chatting entity and the human will be through simple text enabled media like a standard computer terminal, I expect that the general principles arising out of this research will be applicable to all conversations, including voice based conversations. There are additional complexities to voice based conversations, which are beyond the scope of this research, as this work will restrict itself to the analysis of conversational principles and abstract out other dynamics of the conversation process.

One major task of this work is to design an integrated architecture for a chatter bot. The architecture should enable the chatter bot go beyond mere question-answer exchanges and be able to hold a short conversation. It is assumed that the chatter bot will be deployed to provide customer service, and will hold conversations with existing and potential customers. It will answer FAQ type questions, try to resolve customer service issues, try to spot opportunities during the conversation to disseminate unsolicited information, and be able to adjudge the semantic flow of a conversation. If the flow of the conversation requires the chatter bot to pursue a course of action beyond its programmatic capabilities, it will be able to realize this and transfer the conversation to a human representative. The validation for these will be based on Grice's cooperative maxims of conversations [30, 31, 32, 51].

Thus, I list two specific hypotheses for this work.

* My knowledge representation framework will offer sufficient representational power for the task of characterizing information and relationships in the domain of customer service background knowledge.
* My semantic conversation control algorithm will be able to detect transitions in the states of the conversations, predict probable outcomes of the conversation, and use this knowledge to control the conversation.

1.5 Problem Scope

While the proposed research borrows ideas from several other works, it solves a distinct problem. In this subsection, I want to make a clear distinction between the class of problems that this work explores and the class of problems addressed by other well defined areas of research.

Natural Language Processing (NLP) and Computational Linguistics are branches of research that look at the interaction between computing platforms and human languages. There is currently a heavy emphasis on statistical modeling of linguistic artifacts like grammar, parts of speech, and syntactic structures specific to that language.

Some specific problems explored by these fields of research are automatic summarization of a large chunk of text, resolution of reference or objects by several words in a sentence, analysis the natural discourse relationships between sentences, recognizing and classifying the speech acts in a chunk of text. Other NLP research involves automatically translating text from one human language to another, separating words into individual morphemes and identifying the class of the morphemes. Related research areas include performing named entity recognition, converting information from computer databases into readable human language, converting chunks of text into more formal representations such as first-order logic structures that are easier for computer programs to manipulate, identifying parts of speech in a sentence, and performing grammatical correctness analysis of a sentence using language-specific parse trees.

The common theme among all these application problems is that the analysis happens on the sentence level. Each sentence is considered a unit of data to be analyzed. Grammar, parts of speech, reference of pronouns, etc are important features to be studied [47, 39].

However, in my research, I am not concerned with low level abstractions like sentences, phrases, and words. I am taking artifacts like grammar, discourse resolution, and parts of speech to be given. For my proposed research, conversations are the unit of analysis. Each conversation, comprising a series of exchanges, is a data point. I am analyzing the conversation and not the individual components of the conversation at a lower level of granularity. I am trying to determine progression of a conversation over several states, probability of traversing between these states, and exploiting a stored knowledge base to seed the conversation. I want to make a clear distinction between analyzing whole segments of conversations as data units in my proposed work, and analyzing individual sentences, grammatical minutiae and precise problems of discourse resolution as data units in tradition computational linguistics research.

Topic Modeling is a branch of machine learning research that studies statistical models for discovering the abstract *topics* that occur in a collection of documents, most commonly using latent semantic analysis, latent Dirichlet allocation and supervised and unsupervised clustering [87]. While I will borrow these concepts for my knowledge capturing task, my work will go several steps further. The topic modeling techniques are used to drive the knowledge engine for the chatter bot as explained in Chapter 4.

Sentiment analysis research tries to determine favorability / unfavorability of sentences, degree of acceptance of themes, opinion and buzz regarding certain topics, etc [64]. While I will be exploring the idea of changing sentiment over time, I will be looking at sentiments in terms of transition of the conversation over several phases represented by a state-based process.

Thus, in summary, these are the major questions that this research aims to answer.

1. What is an artificial conversation? While this chapter illustrated several examples of artificial conversations, both in general purpose and domain specific situations, it would be useful to develop a principled definition.

2. What does it mean to have better or higher quality conversations? It might be obvious merely from reading the transcript of a conversation whether it is a good conversation or a bad conversation. To develop a principled mechanism of generating higher quality artificial conversations, it is necessary to develop a principled notion of good and bad.

3. What are consistent and principled metrics with which to evaluate artificial conversations? Will these apply to general purpose conversations only or also to domain specific conversations like virtual customer service agents?

4. What are the general design principles and tools that can be used to generate artificial conversations in all domains?

5. What are the general design principles and tools that can be used for knowledge representation needed to support artificial conversations?

6. What are the general design principles and tools that can be used to model the semantic complexity of higher quality artificial conversations?

7. What are the general design principles required to combine the content modeled by knowledge representation and the semantics necessary to ensure the human-like natural quality of artificial conversations?

The rest of the document explores all these questions. Chapter 2 explores existing literature in the area of conversation engineering, focusing on linguistic, statistical, and semantic approaches. Chapter 3 talks about specific approaches in literature for knowledge representation and semantic modeling in general, not specifically applied to the problem of artificial conversations. Chapter 4 explores the traditional scientific metrics used for conversations in general, not necessarily artificial conversations. This chapter will also develop metrics used to evaluate the artificial conversations produced by this work. Chapter 5 will describe the architecture designed to achieve the goals of this research, i.e., generate higher quality artificial conversations. Chapter 6 will talk about the generation of artificial conversations using the chatter bot architecture and describe all the steps in detail. Chapter 7 talks about results and their analysis and discussion. Chapter 8 talks about conclusion, some possible applications of this work, and suggests some future directions.

The unique contribution of this dissertation is demonstrating a novel architectural design to combine content semantics with pragmatic semantics to generate high quality artificial conversations in a specific domain and a narrow situational context. A set of relevant evaluation criteria will be defined and the artificial conversations will be graded according to them. This work will borrow ideas from the literature on conversation theory, knowledge representation, and the theory of pragmatics. This dissertation is the first example of computational modeling of concepts from these fields of research.

Chapter 2

Conversation Engineering

Chapter 2 discusses several studies in the literature that try to solve problems related to the task in this dissertation. These studies can be roughly classified into three groups: the syntactic approach, the stochastic approach, and the semantic approach. Stochastic approaches use statistical frameworks like Bayesian theory, Hidden Markov models, and n-gram modeling to construct individual sentences. Syntactic approaches use techniques from natural language processing and computational linguistics to parse and model grammatical constructs of conversations. Semantic approaches use models of meaning from a knowledge structure to drive sentence analysis and construction. The most representative studies in each approach are described here in detail. As with most research, studies under all three umbrellas heavily leverage ideas from each other.

2.1 The Stochastic Approach

The earliest formal architecture designed for conversations was the *Bayesian Receptionist* at Microsoft Inc [36]. The system maintained a domain of dialogues about goals typically handled by receptionists at the front desks of the buildings on the Microsoft corporate campus. The system employed a set of Bayesian user models to interpret the goals of speakers given evidence gleaned from a natural language parse of their utterances. Beyond linguistic features, the domain models took contextual evidence into consideration, including visual findings.

The 3-level task abstraction hierarchy was the key feature of the system. Each level modeled a different level of abstraction. Level 0, the highest level of abstraction, modeled the task of discriminating the high-level goal of the user, given initial observations and an initial utterance. Level 1, the next lower level of abstraction, modeled the refinement of the high-level goals into more specific goals. Level 2, the lowest level of abstraction, modeled additional conditions for specific situational cases. Levels more detailed than the highest level included an additional state representing the proposition that the current level is inappropriate.

Inference about the belief assigned to each state was used to control backtracking in conversation. Decomposition of a user's goals into several levels of detail allowed for guiding conversation on a natural path of convergence toward shared understanding at progressively greater levels of detail. Multiple levels also allowed for the establishment of common ground [10] about uncertainties at each level. It also allowed conversation regarding comprehension of misunderstandings before progressing to the next level of detail.

In the Microsoft system, users could be directed, as part of a natural dialog about their goals, to implicitly or explicitly confirm or disconfirm misunderstanding at each level. This lead to reduction of uncertainty to some tolerable level before progressing to the next level. The limitation of this model was that it scaled poorly. It was unable to model any conversation beyond asking for direction. But, it also introduced some important ideas about leveraging probabilistic methods for inferencing at different levels of abstraction.

Paek and Horovitz [62] then demonstrated how conversations could be modeled as an inference and decision making problem under uncertainty. They designed *Quartet*, a task independent, multimodal architecture for supporting robust continuous spoken dialog. Their model had four levels of independent analysis. A channel level established mere exchange of utterances. A signal level established intent. An intention level which modeled semantics of the conversation. Finally a conversation level, modeled a tangible activity or behavior based on the conversation.

The Paek and Horovitz model also included learning the parameters of the inference model using an expectation maximization type algorithm. The architecture maintained a probability distribution function over communication failure modes, and minimized this failure function at each level [15]. One limitation of this model was that it relied on ad hoc policies to deal with failures at each level, and these policies had to be designed independently for every class of conversations.

Li and Ji [42] used a probabilistic framework based on dynamic Bayesian networks to model an user's affective states. Although they model general behaviors, not conversations, they were one of the earlier works that introduced the concept of state-based modeling for human behavior. They systematically modeled the uncertainty, dynamics, and different types of knowledge associated with user affective state using DBNs. They demonstrated an information-theoretic mechanism to perform active behavior learning. They also demonstrated user affective state inference in a timely and efficient manner, and proposed information-theoretic criteria to optimally determine when and what assistance to provide to maximize the chance for returning the user to its normal state while minimizing interference with the user's workload and annoyance. The main contribution lies in the integration of the proposed active sensing mechanism into the DBN-based framework for user state inference and user assistance.

Mishne et al. [53] introduced a novel method of estimating the domain-specific importance of conversation fragments from call center telephone calls, based on divergence of corpus statistics. The main novelty of their system was a method for identifying the domain-specific importance levels of fragments in the call, and usage of this method for retrieving possible solutions to the problem presented in the conversation, and for detecting abuse of the call-center resources. A simple way of estimating the significance level of a fragment of a call is to estimate the significance level of each word in the fragment and combine these individual values. In most text analysis systems, the significance of words is inversely related to their frequency. The more common a word is, the less significance it has. Estimating the significance level of a word requires an evaluation of how characteristic the word is to a specific domain, compared to other domains. Rather than global significance, they actually estimated domain-specific word significance (and hence, domain-specific fragment significance). They also used manually transcribed data for validating their results.

Ozyurt and Kose [61] used Naive Bayes, k-nearest neighbor, and support vector machines to automatically mine chat conversation topics in Turkish language call center conversations. Threads and endings of the topics were determined by making analysis at the sentence level rather than the conversation level. They used a Dirichlet prior distribution to initially model the distribution of topics across each sentence, and then support vector machines to learn their final distributions.

Douglas et al. [17] designed a tool for mining customer care chats for news items of importance. Relevant business and dialog features were extracted from the speech logs of caller-system interactions and tracked by a trend analysis algorithm. Features that moved outside their expected bounds on a given day generated headlines as part of a web site generated completely automatically from each day's logs.

2.2 The Syntactic Approach

Many researchers have tried to solve the specific problem of mining and modeling conversations in the context of customer service agents using a variety of cookbook techniques. I will describe the most prominent approaches.

Takeuchi et al. [91], designed a method to analyze transcripts of conversations between customers and human agents at a service center. Their aim was to obtain actionable insights from the conversations to improve agent performance using a three step approach. First, they segmented the call into logical parts. Next they extracted relevant phrases within different segments. Finally, they performed 2-dimensional association analysis to identify actionable trends. They used real conversational data from a service center to identify specific actions by agents that resulted in positive outcomes. They showed that associations between agent utterances and outcomes could be found by segmenting the calls and aggregating phrases within the call segments. In particular they analyzed calls to answer some of the key questions faced by question-answers with the objective of improving agent productivity.

In subsequent work, Takeuchi et al. [92] argued that the language patterns in the final stages of electronic negotiations are more indicative of the outcomes, whereas in face-to-face negotiations the initial stages of the negotiations are more useful for predicting the outcome.

Roy and Venkata. [74] used unsupervised learning algorithms to generate domain models automatically from telephone conversation transcriptions. Their domain model was comprised primarily of a topic taxonomy where every node was characterized by topics, typical questions and answers, typical actions, and call statistics. This hierarchical domain model contained summarized topic specific details for topics of different granularity. However they did not use any semantic features in their taxonomy.

2.3 The Semantic Approach

Chai et al. [5] introduced a fine-grained semantic model that characterized the meanings of user inputs and the overall conversation from multiple dimensions for unified multi-modal input understanding. They also realized discourse interpretation through an integrated interpretation approach that identified the semantics of user inputs and the overall conversation using a wide variety of contexts. They achieved a 90 % accuracy in recognizing the meaning of user input. However the interpretation rules were manually contracted for the models. The rules had to handcrafted each time for different domains, and as a result scaled poorly.

Mehta and Corradini [49] demonstrated the representational power of ontologies for a spoken dialog system. Their system focused on the categorization of ontological resources in to domain independent, and domain specific components. These domains were leveraged to augment the agents conversational capabilities and enhance the systems reusability across conversational domains. They leveraged Google directory categorization for a semi-automatic understanding of a user utterance on general purpose topics like movies and games.

2.4 Dialogue systems

There has been a long history of research in dialogue systems, both speech-based and text-based. Some dialogue systems also incorporate graphics, haptics, gestures, and other mediums of communication. These systems have been deployed in a wide range of applications like website navigation assistance, virtual receptionist, training and tutorials, virtual concierge at hotels, etc.

One of the earliest research efforts was the GUS (Genial Understander System) [3]. This was a virtual agent helping a customer make reservations. The knowledge base about flights options, timings, costs, etc. were encoded in the form of frames. Frames were used to represent collections of information at many levels within the system. Some frames described the sequence of a normal dialog, others represented the attributes of a date, a trip plan, or a traveller. While the system worked well on handling reservations, it wasn't particularly intelligent. It could handle only a very restricted set of questions, and the domain knowledge of the question-answer sequence had be encoded exactly in the frame in the same order in which the questions would be asked.

The GALAXY Communicator system at MIT [81, 68] is a client-server architecture for communicating online information like weather and flight information. It has several components like database access, speech synthesizer, speech recognizer, and a language understanding engine. It has achieved good results in travel reservation domain, and is available as an API to build an end to end system [68]. The knowledge is represented in terms of frames and it uses a turn management system for dialog control. It can handle a very sophisticated range of conversations ranging from yes-no questions to answering complex queries [19]. It's main limitation is that it is not set up to build the knowledge base using facts, but in terms of anticipated questions [19].

FASIL [14] was an email summarization system for use in a voice-based virtual personal assistant. It used an XML-based dialogue manager that used named-entity recognition. It was optimized for email summarization, as email has distinct characteristics as compared other forms of text.

The DARPA Communicator project [41] was an initiative to support advanced conversational capabilities including negotiation, plan optimization, and complex explanations. All the above mentioned dialog systems were part of a DARPA initiative. Some metrics used to evaluate the system were the number of error messages, the mean system turn duration, the mean user turn duration, the number of system words to task end, the number of user words to task end, the mean response latency, and the total duration of task [96].

State tracking is an important task in management of dialog systems. Several belief based state tracking architectures handle this problem using stochastic methods. These include generative and discriminative models [16]. Some specialized techniques leverage dialogue structure in specific context to improve accuracy by encoding speech recognition patterns [50]. Neural networks have also been used for deep-learning solutions to this problem [35].

Partially Observable Markov Decisions Processes (POMDPs) have also been used to model conversations. They improve upon traditional conversational systems in that they can better handle ambiguity from changing domains [25]. Reinforcement learning techniques have also been used for this problem. [73]

2.5 Limitations of existing approaches

Although there has been a lot of progress made over the years in the design of conversational engineering systems, there is one major limitation to most of them. They do not make an explicit distinction to modeling the content required for the conversation and the semantics inherent in the conversation process.

Most approaches either focus on just one of content modeling or conversation semantics, or sub-aspects of these, or incorporate both of them together without making an explicit distinction. This leads to blind spots in the application, in which either one has to encode content and semantics to for a new domain from scratch, or the system has to undergo substantial remodeling to handle conversations of a different type.

This dissertation is based on the assumption (as shown in the next chapter) that a conversation is a specific well-defined process, that follows certain explicit conventions. These conventions almost always include content and semantics, and in some cases may include additional features like sentiment and emotions. The next chapter, which evaluates literature from the theory of conversation analysis, not necessarily artificial conversations will strengthen the case for the assumption that content and semantics are distinct features that need to be modeled separately and then made to work in conjunction to generate artificial conversations.

Chapter 3

Modeling Content and Semantics

This chapter explores the problem of representing a conversation and the background knowledge engine. One key aspect distinguishing this research from existing approaches is that I consider conversations to the unit of my analysis. I am modeling various aspects of the conversation itself, without delving into lower-level grammatical and linguistic minutiae.

3.1 Analysis of Conversations

What is a conversation? There is no scientifically accepted definition, but the literature enumerates several characteristics in general.

* Conversations are based on a series of reactions to previous utterances. They have an interactive structure [26, 83].
* Conversations may be unpredictable, but they are rarely random. They tend to exhibit some semi-structural characteristics based on established social conventions. For example, a conversation between two or more friends may be

spontaneous. A conversation between a doctor and a patient may be more structured. A conversation between an interrogator and an interrogatee may be very one sided with only the interrogator making most of the utterances [26, 78].

* Conversations will often have a shared goal, towards which all participants of the conversation try to proceed. The goal can be subjective, for example, to agree on some point of view, or objective, for example, to negotiate to a specific price point. The goal may shift over the course of the conversation. There may not be any goal in case of idle gossip among friends. Small talk too has a goal, that of increasing comfort level of the conversation participants [82, 83].

* Conversations are sometimes the ideal form of communication in customer service situations, depending on the participants' desired end goals. Conversations may be ideal when, for example, each party desires a relatively equal exchange of information, or when one party desires to query the other to solicit information [78, 82].

3.1.1 Structure in Conversations

Conversation analysis is the study of talking interactions, in situations of everyday life and is considered an important subfield of research by linguists, anthropologists, and sociologists. It has been used as a tool for research in interactional sociolinguistics, discourse analysis, discursive psychology, and phonetics [27, 46]. Basically, the goal of conversation analysis is to find interesting patterns, structure, and order in specific situational contexts like a classroom, a court of law, or a doctor's office [22].

Some basic structures of a conversation according to conversation analysis are as follows.

* Turn-taking is often the most fundamental structure observed in conversations. In a conversation between two individuals, each individual takes turns speaking. The duration of the turns may be uneven, but they do take turns. Literature defines this as having two sub-processes: allocation of turns and construction of turns. Allocation of turns describes how the turns of individual speakers (two or more) in a conversation is organized, either explicitly or implicitly. Turn construction describes lexical, clausal, phrasal, and sentential conceptual units that make up the turn [26, 82, 83].

* Conversations in which the participants are mostly agreeing with each other may have a different structure than conversations in which the participants are disagreeing with each other [70].

* Often conversations tend to drift away from the intentions of the participants. This may be due to issues related to miscommunication, or due to the contents of the conversation. Consequently, the participants may attempt to bring the conversation back on course to align with intentions. This is known as the process of conversation repair. The repair mechanism can be explicit or more subtle, but it has some order and structure to it. This structure may depend on social or institutional conventions [75, 78].

Gordon Pask's conversation theory [67] describes how shared knowledge is constructed through conversation. This is dependent on the content of the conversation, the behavior of the participants of the conversation, and the mutually agreed meaning of the contents as defined by some social or institutional convention. This process may be further dependent on narrow situational contexts, for example, a conversation between a teacher and a student in a classroom [67]

3.1.2 Speech Act Theory

Speech act theory asserts that with each utterance in a conversation, an action is performed by the speaker [101, 102]. These actions, related to the utterance, can be classified into several different profiles [95]. These profiles can be put together into predefined patterns to constitute an entire conversation [28]. Winograd and Flores [101, 102] show that conversation for action is an example of a pattern of speech acts organized together to create a specific type of conversation [95]. Speech act modeling can also be applied to logs of stored conversations by manual or automated annotation of individual utterances [95]. The modified SWBD-DAMSL tag set [38] is a comprehensive collection of all speech acts defined in the literature. These can be modeled using hidden Markov models [72].

Speech acts can be broadly classified under three levels of language-action associations. [2, 1, 90].

1. **Locutionary:** This refers to the actual utterance and its intended meaning, where the utterance is meant to be taken completely literally, with no consideration for context or conventions.

2. **Illocutionary:** This refers to the utterance and its real intended meaning, i.e., what utterance actually means in a particular context as defined by some convention of social linguistic usage.

3. **Perlocutionary:** This refers to the utterance and its possible unintended consequences or effects, i.e., the utterance causes fear, anxiety, some rational or irrational action, etc.

The illocutionary speech act is the most semantically meaningful of the three levels as applicable to this research. The literature on illocutionary speech acts broadly recognizes the following four taxonomies [54].

1. **Austin:** Expositives, Exercitives, Verdictives, Commissives, Behabitives.
2. **Searle:** Representatives, Directives, Commissives, Expressives, Declarations, Representative Declarations.
3. **D'Andrade and Wish:** Expositives, Interrogatives, Exercitives, Reactions, Verdictives, Commissives, Behabitives.
4. **VerbMobil:** Request, Suggest, Convention, Inform, Feedback.

The choice of which taxonomy to use, or which subsets of taxonomies to use depend on the application. It is also possible to mix and match between different taxonomies depending on the application [54].

Dialogue acts are context-specific speech acts that have been defined for specific applications [88]. They can refer to speech acts in a narrow situational context. Some common dialogue acts are: greetings, Meta-Questions, Yes-No Questions, Statements, Requests, Wh-Question.

Just like speech acts, dialogue acts can have taxonomies too. A dialogue act taxonomy must trade-off between two considerations. The definitions of the dialogue act tags must be clear enough in order to be easily distinguishable in the specific context where it's being applied. They must also be general enough to be reusable for other applications as well [20].

3.1.3 Semantics and Sentiment in a Conversation

Sentiment analysis research tries to identify emotions and attitudes from text. For example, trying to determine if a newspaper article has a positive, negative, or neutral connotation. A related area of research, opinion mining, tries to deduce subjective information from pieces of text [64]. These tasks usually use sophisticated algorithms from natural language processing, computational linguistics, and text analytics. The piece of text could be a document, a single sentence, or just a word. Sentiment analysis in conversations tries to deduce the attitude or emotional state of the speaker and how it influences the conversation, i.e., is the speaker angry, sad, happy, unhappy, impatient, etc.

Pang [65] and Turney [94] applied sentiment analysis to film and product reviews to determine if they were favorable or unfavorable. Pang and Lee [66] also analyzed documents for polarity of sentiment on a continuous rather than discrete scale. Snyder [84] applied sentiment analysis to restaurant reviews and used it to predict ratings on a star system for the food, service, and ambience of the restaurant. Feature or aspect-based sentiment analysis is more fine-grained and tries to determine opinion polarity on specific entities mentioned in the text under review [37, 45, 44]. For example, the viewfinder of a digital camera, the service at a bank, etc.

The accuracy of a sentiment analysis system indicates how well the analysis agrees with human judgments. This is usually measured by precision and recall. Pang and Lee [64] showed that human raters typically agree with each other about 70% of the time. Thus 70% is a good benchmark to test sentiment analysis methods. Pang and Lee [64] also claim that correllation is a better measure than precision because it takes into account how close the predicted value is to the target value.

The microblogging platform Twitter (`http://www.twitter.com`) provides an ideal corpus for sentiment analysis. Pak and Paroubek [63] classified tweets as positive negative or neural using a naive-Bayes classifier based on n-grams and a parts-of-speech tagging algorithm.

3.2 Knowledge Representation for Conversations

Conversation are usually based on a specific context. The participants of the conversation usually possess the background knowledge upon which they make their utterances. How is knowledge traditionally represented? This section gives an overview of knowledge representation research, not necessarily for conversations, but in general for modeling any task that requires some background knowledge.

3.2.1 Goal Fulfillment Maps

Text-based chatter bot scripts are implemented using hierarchical rules in the form of scripts [56]. Each rule possesses a list of structural patterns of sentences and an associated response. In some cases, text-based chatter bots are designed to work in a goal-oriented manner with the aim to achieving a specific goal [52]. Since a single chat utterance can be input in many different ways, a very large number of scripts would have to be designed. This is an onerous task [59, 56]. Modification of the rules would be difficult, since all rules might need to be changed for the sake of consistency. The designer has to predict how each change would influence other rules [76]. It has been highlighted that by employing sentence similarity measures, scripting can be reduced to a few prototype sentences [43, 59, 56, 58, 57]. A goal-fulfillment map is an effective way to represent the condensed knowledge in scripts. It is based on the conversation agent semantic framework by O'Shea [56, 58, 57, 60]. The rules are described in such a way that they can be sequentially combined to satisfy some goal. The user must traverse some contexts to achieve the goal.

O'Shea [56, 58, 57, 60] explains that in a conversational context, "the contexts along the goal-fulfillment map express specific queries, which require specific answers in order for progression to be made along one of possibly various routes. Users may switch between contexts, traversing forwards or backwards along the goal-fulfillment map. Engaging in dialogue with a user, the chatter bot is able to capture specific pieces of information from the user input in order to progress along the network of contexts described by the goal-fulfillment map. Using a goal-oriented goal-fulfillment map, the aim is to elicit a specific set of answers from the user in order to achieve goal-fulfillment. As such, rules that pose a question possess a series of successor rules, which seek to obtain an appropriate answer. If no relevant input is sent, that is, no match found above that of a preset threshold, an associated default rule is returned prompting the user for further input to satisfy the request."

Figure 3.1: Building blocks of a goal-fulfillment map [58, 57, 60].

Each rule is assigned an activation level of one. Once a rule has fired its activation level is reduced to zero and as such will fail to fire again from subsequent user input. For example, if a rule has performed its task of eliciting a piece of information from the user, its purpose is no longer required. Figure 3.1 shows the building blocks of a Goal Fulfillment Map [58, 57, 60].

The selected domain as described by the goal-fulfillment map in figure 3.2 is concerned with advising credit card customers on debt management. The goal-fulfillment map contains seven interconnected contexts. The contexts along the goal-fulfillment map express specific queries, which require specific answers in order for progression to be made along the designated route.

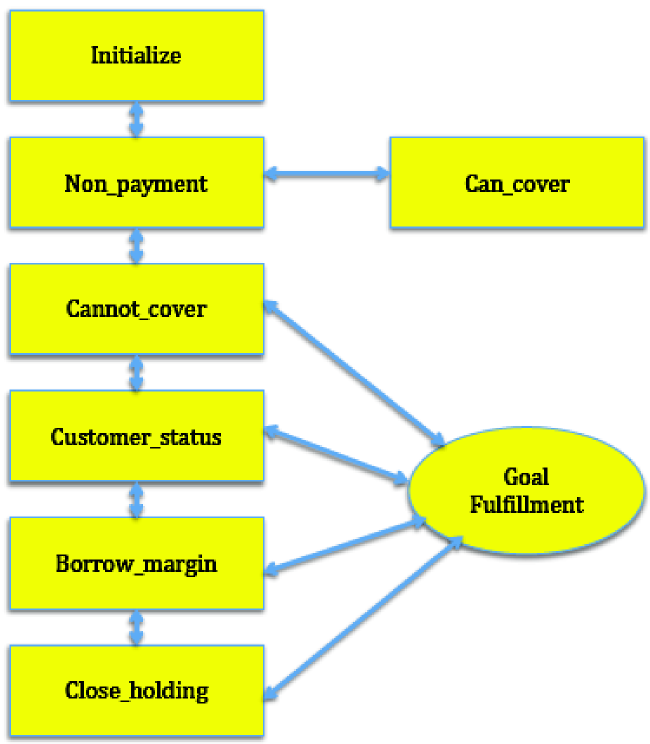

Figure 3.2: A Goal Fulfillment Map [58, 57, 60]: where a specific map encodes the sequence of questions to be asked, whose responses will enable fulfill a specific goal during the conversation.

Dialogue will traverse the goal-fulfillment map in a progression starting with the base context named *Initialize*. It is possible to revert back to a previously visited context in the case of a misinterpreted line of input. As such, the user has the option to alert the chatter bot that there has been a misunderstanding.

The following contexts are represented.

* *Non Payment* aims to elicit the reason for non-payment of the debt;

* *Can Cover* identifies that the customer has in fact paid the debt and as such reached goal-fulfillment;

* *Cannot Cover* aims to elicit why a customer simply cannot afford even to pay the proposed minimum payment of the debt;

* *Customer Status* identifies the status of the customer, for example,

    * *Regular Customer* has fewer privileges and options to defer minimum payment.

    * *Elite Customer* has more options to defer minimum payment.

* *Borrow Margin* determines whether the customer has applied for borrowing margin from another account.

* *Close Holding* aims to identify if the Customer's account should be closed or suspended due to non-payment of debt.

* *Default* context is one in which the chatter bot will revert to if no content within the current context is appropriate to that of the user input.

3.2.2 Ontologies

The formal study of ontologies has its roots in the philosophical and metaphysical study of nature. An ontology can be considered to be a formally rigorous description of a specific domain of knowledge. Philosophers consider this to be a systematic

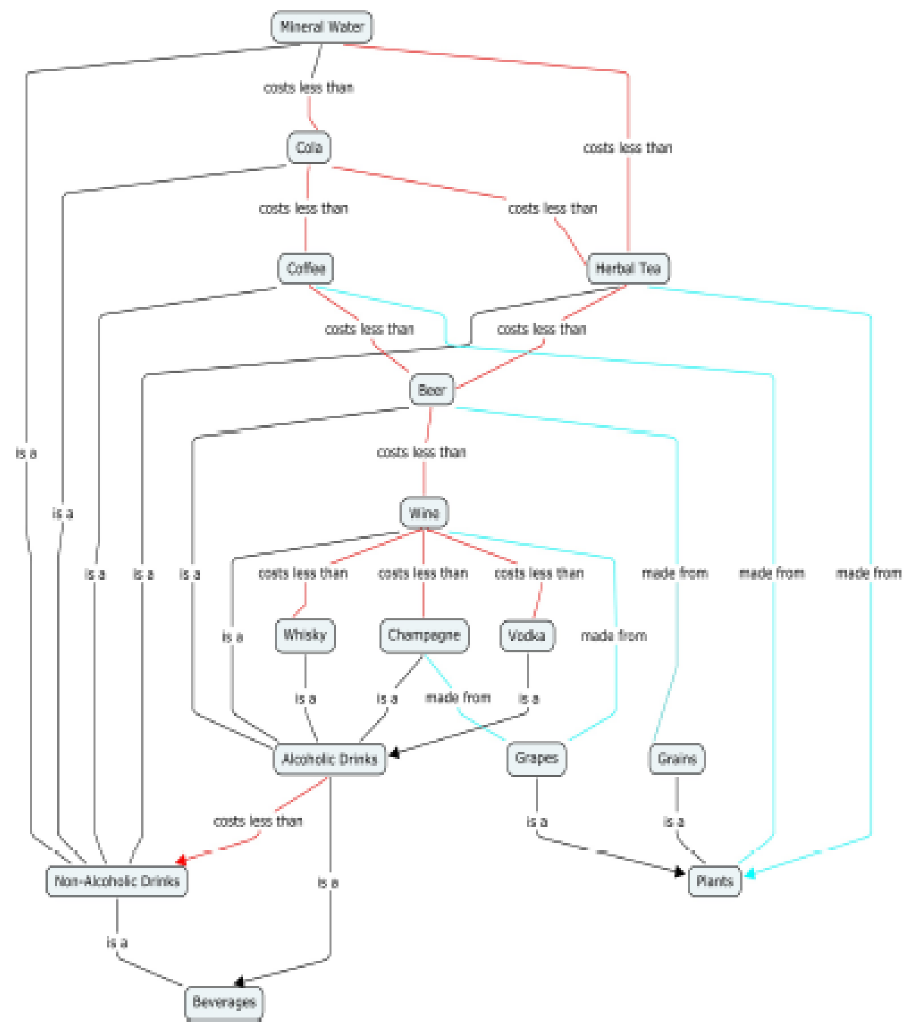

Figure 3.3: A Wine Ontology: represents knowledge about beverages.

Acknowledgement: Jessica Turner & Matthew Turner, Mind Research Network.

account of existence, because in a knowledge-based system, that which exists can be represented. According to Gruber [33], "when the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects, and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge. Thus, we can describe the ontology of a program by defining a set of representational terms. In such an ontology, definitions associate the names of entities in the universe of discourse (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names are meant to denote, and formal axioms that constrain the interpretation and well-formed use of these terms" [33].

Ontology addresses the science of the most general properties of things, which includes both the properties of the thing in question, and also the properties of every a priori determined thing. In the field of information organization and extraction, an ontology is a formal representation of knowledge as a set of concepts within a domain, and the relationships between those concepts. It is used to reason about the entities within that domain, and may be used to describe the domain. Gruber [33] defines an ontology as a "formal, explicit specification of a shared conceptualization". An ontology provides a shared vocabulary, which can be used to model a domain, i.e., the type of objects and/or concepts that exist, and their properties and relations.

A domain ontology describes a specific area of focus. Figure 3.3 shows a very simple domain ontology representing knowledge about beverages. The atomic instances are Mineral Water, Cola, Coffee, Herbal Tea, Beer, Wine, Whisky, Champagne, Vodka, Grains and Grapes, which are individual well-grounded concepts. The classes, which are collections of individual concepts and sub-classes, are Alcoholic Drinks, Non-alcoholic Drinks, Beverages and Plants. The relationships are *is a*, *costs less than*, and *made from*.

The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies. The languages are characterized by formal semantics and RDF/XML-based serializations for the Semantic Web. OWL is endorsed by the World Wide Web Consortium (W3C) and has attracted academic, medical and commercial interest.

Similarly, conceptual graphs are a system of logic based on the existential graphs and the semantic networks of artificial intelligence [85]. They express meaning in a form that is logically precise, humanly readable, and computationally tractable. With their direct mapping to language, conceptual graphs serve as an intermediate language for translating computer-oriented formalisms to and from natural languages [29]. With their graphic representation, they serve as a readable, but formal design and specification language.

Conceptual graphs have been implemented in a variety of projects for information retrieval, database design, expert systems, and natural language processing. [69, 86] According to Obitko [55], "although the formalism for conceptual graphs is based on semantic networks, it has direct translation to the language of first order predicate logic, from which it takes its semantics. The main feature is standardized graphical representation that like in the case of semantic networks allows a human to get a quick overview of what the graph means. A conceptual graph is a bipartite orientated graph where instances of concepts are displayed as a rectangle and conceptual relations are displayed as an ellipse. Oriented edges then link these vertices and denote the existence and orientation of relation. A relation can have more than one edge, in which case edges are numbered. Conceptual graphs have the same expressing power as predicate logic"

Basically, a conceptual graph is a bipartite graph containing two kinds of nodes, concepts and conceptual relations. Arcs link concepts to conceptual relations, and each arc is said to belong to a conceptual relation. There are no arcs that link concepts to concepts or relations to relations. Concepts have an associated type and a referent. A referent is a way to denote the entity of the universe of discourse to which the concept refers. It consists of a quantifier, a designator, which either points to the referent or describes it, and a descriptor, which is a conceptual graph describing some aspects of the referent. Note that a quantifier here may only be of existential kind or specify that a precise number of instances of the referent exist. A descriptor is considered as a conceptual graph nested in to a concept. The concept is said to be a context for the nested conceptual graph. Conceptual relations are are also typed. A relation type associates to a conceptual relation a valence equal to the number of arcs that belong to the relation, and a signature that constraint the types of concepts linked to the relation [29, 69, 85, 86].

Chapter 4

Evaluation of Conversations

This chapter describes the theoretical background employed to validate the results of my research. It is important to have a standard and consistent metric to measure the quality of conversations. I borrow ideas from the theory of pragmatics to define some metrics to evaluate conversations.

4.1 Theory of Pragmatics

Pragmatics is a subfield of linguistics which studies the ways in which context contributes to meaning. Pragmatics encompasses speech act theory, conversational implicature, talk in interaction and other approaches to language behavior in philosophy, sociology, and linguistics [51]. It studies how the transmission of meaning depends not only on the linguistic knowledge (for example, grammar, lexicon, etc.) of the speaker and listener, but also on the context of the utterance, knowledge about the status of those involved, and the inferred intent of the speaker. In this respect, pragmatics explains how language users are able to overcome apparent ambiguity, since meaning relies on the manner, place, time, etc. of an utterance.

Pragmatics is a systematic way of explaining language use in context. It seeks to explain aspects of meaning which cannot be found in the plain sense of words or structures, as explained by semantics. As a field of language study, pragmatics is fairly new. Its origins lie in philosophy of language and the American philosophical school of pragmatism. As a discipline within language science, its roots lie in the work of Paul Grice on conversational implicature and the cooperative principles [30, 31, 32, 51].

The cooperative principle describes how people interact with one another. As phrased by Grice, who introduced it, "Make your contribution such as it is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged." Though phrased as a prescriptive command, the principle is intended as a description of how people normally behave in conversation.

4.1.1 Grice's Maxims

The cooperative principle can be divided into four maxims, called the Gricean maxims, describing specific rational principles observed by people who obey the cooperative principle that enable effective communication. Grice proposed four conversational maxims that arise from the pragmatics of natural language. The Gricean Maxims are a way to explain the link between utterances and what is understood from them [30, 31, 32, 51].

Grice proposes that in ordinary conversation, speakers and hearers share a cooperative principle. Speakers shape their utterances to be understood by hearers. Grice analyzes cooperation as involving four maxims: quantity, quality, relation, and manner. Speakers give enough and not too much information (quantity maxim). They are genuine and sincere, speaking "truth" or facts (quality maxim). Utterances are relative to the context of the speech (relation maxim). Speakers try to present meaning clearly and concisely, avoiding ambiguity (manner maxim).

Grice's cooperative principles are a set of norms expected in conversation. Grice's maxims of cooperation can also be interpreted in conversations as follows:

* quality: speaker tells the truth or provable by adequate evidence
* quanity: speaker is as informative as required
* relation: response is relevant to topic of discussion
* manner: speaker's avoids ambiguity or obscurity, is direct and straightforward

Saygin et al. [77] demonstrated that evaluating chatter bots using Grice's cooperative maxims is an effective way to compare chatter bots competing for the Loebner prize. The maxims provide a scoring matrix, against which each artificial conversations can be graded for a specific criterion. Thus this is a good potential starting point for evaluating the artificial conversations in this dissertation.

Since Grice's maxims were proposed to initially evaluate human conversations long before the advent of chatter bots, the chatter bot generated artificial conversations in this dissertation will also be evaluated against this criterion. This will form the basis of comparison of chatter bot performance, i.e., how human-like was the performance of the chatter bot in engaging the customer in a service related conversation.

4.2 Domain and Situation Specific Conversations

The previous section showed that Grice's maxims can be a good "proxy" metric for evaluating artificial conversations, the question arises: given that most feasible artificial conversations will take place in the domain or situation specific context, as shown by Figure 1.2, are there other metrics of evaluation that can be defined? The answer to this question will again depend on the domain. This research aims to produce high quality artificial conversations in the customer-service domain. What are the nature of these conversations that can give some idea about appropriate evaluation metrics?

In customer service situations, a customer has a conversation with a chatter bot via text chat, describes some issue, or seeks some information or guidance, and the chatter bot helps resolve it. The issue is usually complicated enough that it can't be resolved in a single utterance-response exchange. Typically, the conversation will have to go through a few utterance-response exchanges to fully address the issue. Then a few more utterance-response exchanges may be required to carry out the task of resolving the issue.

In order to comprehend a specific issue, the chatter bot must often ask a set of follow up questions. The specific question would be completely dependent on the situational context in the domain. But for a well-defined context, the number of such followup questions will be fixed. For example, if the issue being discussed by the customer has to do with incorrect allocation of margins in a financial account, then to fully comprehend the issue, the chatter bot needs to know if the account is a saving account or a trading account, what is the specific configuration of the account, is the account set up for day trading or regular trading, and what is the minimum margin required by the account. Hence, a good artificial conversation would be one in which the chatter bot ask all or most of these followup questions.

Similarly, in helping the customer resolve an issue, the chatter bot might have to lead the customer through a series of steps. For example, to change the configuration of the account, the customer might have to change the login password, the transaction password, change allocations, change trading frequencies, or reassign balances to margins. In a good artificial conversation, the chatter bot should ask the customer to perform all these steps in some order. Hence the fraction of follow up questions asked by the chatter bot is an important evaluation metric.

It must also be noted that the ultimate function of a customer service chatter bot is to help the customer resolve some issues. These issues could be simply providing information, guiding the customer through some pre-defined procedure like closing an account, troubleshoot some problems or issues and resolve it, or resolve some dispute or argument or difference of opinion with the customer. In each of these tasks, the ultimate goal is to successfully resolve some issue. Thus, one important evaluation criteria could be how many times the chatter bot is actually able to successfully resolve an issue.

Finally, as it has been mentioned several times, the objective of this dissertation is to ultimately enable chatter bots to generate high quality artificial conversations that go beyond simple question-answer or utterance-exchange pairs to a series of utterance-exchange pairs where the context is maintained throughout. In chapter 1, it was demonstrated that contemporary state-of-the-art chatter bots like Mitsuku and customer service bots can only perform well over single pairs of utterance exchanges. Since this research tries to overcome this limitation, it follows that scoring the number of utterance-exchange pairs over which the artificial conversation can maintain coherence is an important evaluation metric.

In this chapter we discussed several potential evaluation criteria. Some are of an objective nature, which can be easily and unambiguously measured. Some of them are subjective, and would need to evaluated in a consistent and rigorous manner. These factors will be considered in the results and discussions chapters where I describe the experimental methodology that performs these evaluations.

Chapter 5

The Chatter Bot Architecture

As demonstrated in chapter 3, a conversation is a process that is governed by certain protocols, both social and functional. These governing protocols are applicable to all general purpose conversations. In purpose driven conversations like customer service interactions, a more restricted set of protocols applies. As such, chapter 4 demonstrated that a good general purpose conversation usually satisfies the four Gricean maxims of Quantity, Quality, Relation, and Manner. A purpose-driven conversation must satisfy further criteria, which in the case of customer service conversations include but are not limited to the percentage of follow-up questions asked, number of turns of utterance exchanges carried out while adhering to the context, and the number of successful resolutions of customer service issues.

A general-purpose chatter bot that grades well against these criteria must essentially have a strong underlying semantic model. The semantic model must capture the inherent underlying process-driven protocol of conversations. In addition, a specific-purpose customer service chatter can leverage the inductive bias inherent in targeted purpose-driven communication capturing the domain characteristics.

Also demonstrated in chapter 3 was the notion of a knowledge background that is necessary for conversations. For general purpose conversations, knowledge of the nature and relations governing the conversations was an integral part of the process. For purpose-driven customer service conversations, a more comprehensive knowledge base is essential. This would be true for any domain specific conversational chatter bot. This is because a conversation is a process that is strongly rooted in content. A conversation is a means of communication about situations and issues that arise from the characteristics and peculiarities of the underlying content. This poses two related challenges: storing the content, and efficiently retrieving the content. The success of an artificial conversation is strongly dependent of being able to retrieve or access the correct piece of content when required. Hence, an artificial conversation system requires good knowledge engineering to store domain specific content that is easy to access, maintain, and update when necessary.

Informally, we can say that a good chatter bot must know *what to say* as well as *how to say it*. Essentially, *what to say* is defined by content, and *how to say it* is defined by semantics. Artificial conversations thus must combine content and semantics effectively. This chapter describes the system architecture and methodology required to do this in an effective and principled way. The architecture borrows several knowledge representation and conversation analysis ideas that were discussed in chapter 3. The objective of the architecture is to engineer conversations that can score well against the qualitative and quantitative metrics discussed in chapter 4.

5.1 Important Definitions

Throughout chapters, 1, 2, 3, and 4, several terms related to conversations have been discussed informally. Before I begin describing the architecture, it will be useful to state all the definitions and terminology in a formal principled manner. These definitions apply only to the scope of this current work, i.e., text chat between a chatter bot and a human in a turn-taking manner.

- **Utterance:** An utterance is everything that is said by either the chatter bot or the human in a single turn. It could consist of one for more sentences. For simplification, it is assumed that each utterance belongs to a single context. Multiple contexts in the same utterance is beyond the scope of this research.

  For Example, this is an example of one chatter bot utterance. (Acknowledgement: This corpus was provided by Next IT Inc., Spokane, WA.)

  Chatter Bot : `Hello. How can I help you today?`

  This is an example of one human utterance.

  Human : `I am unable to login to my account. I forgot my password.`

  This is an example of an utterance that consists of more than once sentence. (Acknowledgement: This corpus was provided by Next IT Inc., Spokane, WA.)

  Customer : `Yes. I sold USD 750 worth of stock in JFD and along with my previous balance of USD 350, I have sufficient margin in my portfolio. Yet I am unable to process my transactions due to insufficient margin error.`

  This is an example of a pair of utterances. The chatter bot and the human take turns with their utterances. It is based on the the turn-taking model of conversation discussed in chapter 3. (Acknowledgement: This corpus was provided by Next IT Inc., Spokane, WA.)

  Chatter Bot : `Can you please provide your mother's maiden name for verification?`

Customer : `It is Parker.`

- **Conversation:** A series of successive utterance pairs on the same context constitutes a conversation. The number of utterance pairs in the conversation can vary. But the entire series belongs to a single context. A conversation with multiple contexts is beyond the scope of this work.

  For example, the following is a conversation. (Acknowledgement: This corpus was provided by Next IT Inc., Spokane, WA.)

  Chatter Bot : `Hello. How can I help you today?`

  Customer : `I received an quote for home insurance last week. I would like to purchase that policy.`

  Chatter Bot : `All right. Do you remember the quote reference number?`

  Customer : `No. I had printed it out but I left the printout in my car.`

  Chatter Bot : `Not a problem. I can look it up by your last name and date of birth. What is your last name?`

  Customer : `Johnson`

Chatter Bot : `What is your date of birth?`

Customer : `April 21, 1973`

Chatter Bot : `Great. I found your quote. would you like to purchase the policy now?`

- **Corpus:** The corpus for this dissertation is a set of several conversations. Each conversation is between a chatter bot and a human following the turn-taking model. Each conversation constitutes a separate data point in the corpus. Each data point in the corpus is obtained from the same domain. The details of the specific corpus used in this work will be described in chapter 6. (Acknowledgement: This corpus was provided by Next IT Inc., Spokane, WA.)

5.2 System Architecture

The system architecture for the chatter bot consists of three main modules, namely, the Knowledge Engine, the Conversation Engine, and the Chat Interface. The Knowledge Engine handles *what to say*, while the Conversation Engine handles *how to say it*. Thus these two modules handle distinct tasks for the artificial conversation process. The Chat Interface module is the third component of the system architecture. This module is the first layer of interface with the user. It carries out the first stage of pre-processing of the raw chat data. The overview of the architecture is shown in Figure 5.1. The actions of all these modules are explained in this chapter.

The chatter bot architecture has been designed to test the hypothesis that has been developed incrementally in chapter 1, 2, 3, 4. As mentioned during the con-

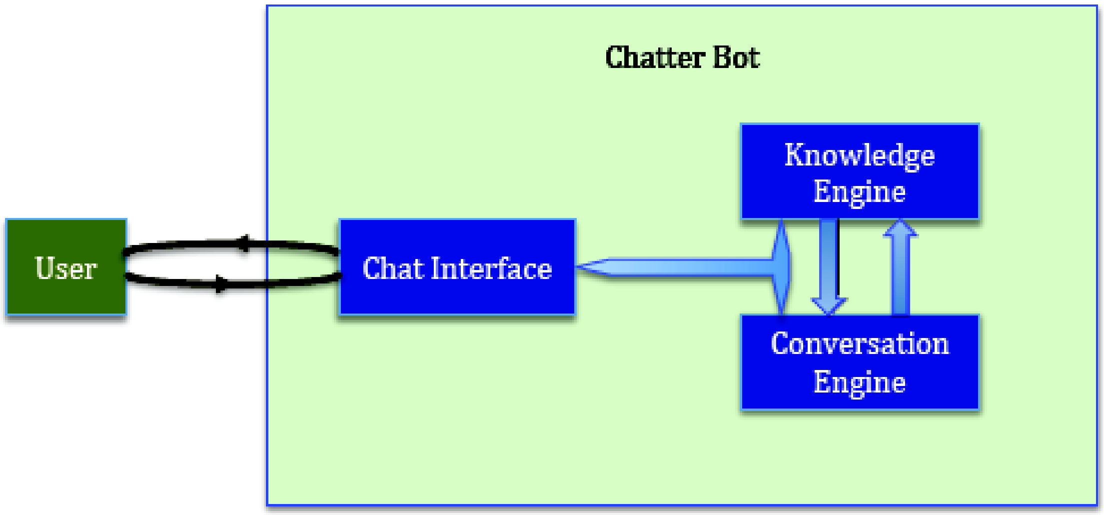

Figure 5.1: System Architecture for the Chatter Bot containing the Chat Interface, which pre-processes the raw chat text, the Knowledge Engine, which provides the content of the conversation, and the Conversation Engine, which manages the semantic context of the conversation. The block arrows indicate direction of flow of information between the interfaces implementing the modules

struction of the hypothesis, the chatter bot needs a knowledge base that drives the content of the conversation. The Knowledge Engine module contains the data structures and methods necessary to fulfill this objective. It was also mentioned during the hypothesis construction that the chatter bot needs to keep track of semantic context during a conversation, and be able to engineer the conversation towards favorable states. This functionality is implemented by the Conversation Engine. The Chat Interface directly connects with the user inputs. In this system, the user, who is the human playing the role of the customer, enters his text-based chat utterances through the standard terminal. The chatter bot produces its response back to the standard terminal. It is a rudimentary system as of now and is not interfaced with a graphical user interface.

The solid blue arrows in the figure show the flow of information. A lot of information is passed between the three modules to get the conversation to work. The remaining sections in this chapter will go into the details. The black bidirectional arrows denote the interaction with the user (human or customer) via the standard terminal, but this could denote a graphical user interaction in a future version. The architecture has been designed to be modular. The Knowledge Engine and the Conversation Engine can be implemented by any of the several concepts described in Chapter 3. In this work they have been realized using specific techniques, which will be described in detail in the next few sections. But the architecture has been designed to be of a plug-and-play nature, where other techniques for representing content and semantics can be substituted in and out to test and experiment with them.

5.3 Knowledge Engine

The Knowledge Engine is the component of the chatter bot architecture that supplies the content of the conversation. The design of the Knowledge Engine is extremely domain dependent. In chapter 3, we discussed two main ways knowledge can be represented: Goal-fulfillment maps and Ontologies.

The two main content defining characteristics of the conversation are the domain-specific information about the subject matter being discussed in the conversation, and the particular speech act being adhered to in the conversation. The former provides the meat-and-potatoes details about the conversation, i.e., the specific information that the customer, or the agent is seeking or providing in the conversation. The latter provides the situational context of the conversation, i.e., what are the qualifiers of the information exchange in the conversation, that determines the conversation engineering strategy of the chatter bot architecture. Without the former, the conversation wouldn't be grounded and would constitute just an abstract process. Without the latter, the conversation would resemble the flawed conversations we discussed in chapter 1. Thus, the combination of both of these defining characteristics is essential to the conversation engineering process.

The next design question is the organization of these two components. In adherence to our modular design principle for maintainability, the information organization is optimized for ease of retrieval and update. Essentially, the algorithmic capabilities of a hash-map data structure is leveraged, i.e., the constant time access, insertion, and deletion, and the linear time space complexity. In this implementation, the standard Java hash map as defined by the collections API is used. Specific hash functions can be leveraged for optimization, but that is currently beyond the scope of this work.

The subject matter information is organized in the form of a Topic Hash Map. The speech act information is organized in the form of a Speech Act Identifier. These hash map implementations are described in detail below, as they are essential for the realization of the engineering goals of this work. The block diagram for the Knowledge Engine is shown in Figure 5.2.

5.3.1 Speech Act Identifier

As explained in chapter 3, there are several differing specifications of speech acts theory from the literature on linguistics and dialog systems. While a speech act is an utterance with a general performative function, and can range from 5 to 10 in number, a dialog act is a specialized speech act that is defined only the context of specific dialog system. All the most commonly and universally defined 42 dialog acts have been collect in the in the modified SWBD-DAMSL tag set [38].

From this collection of both speech and dialog acts, I have identified a specific set of Speech Acts that are appropriate for this work.

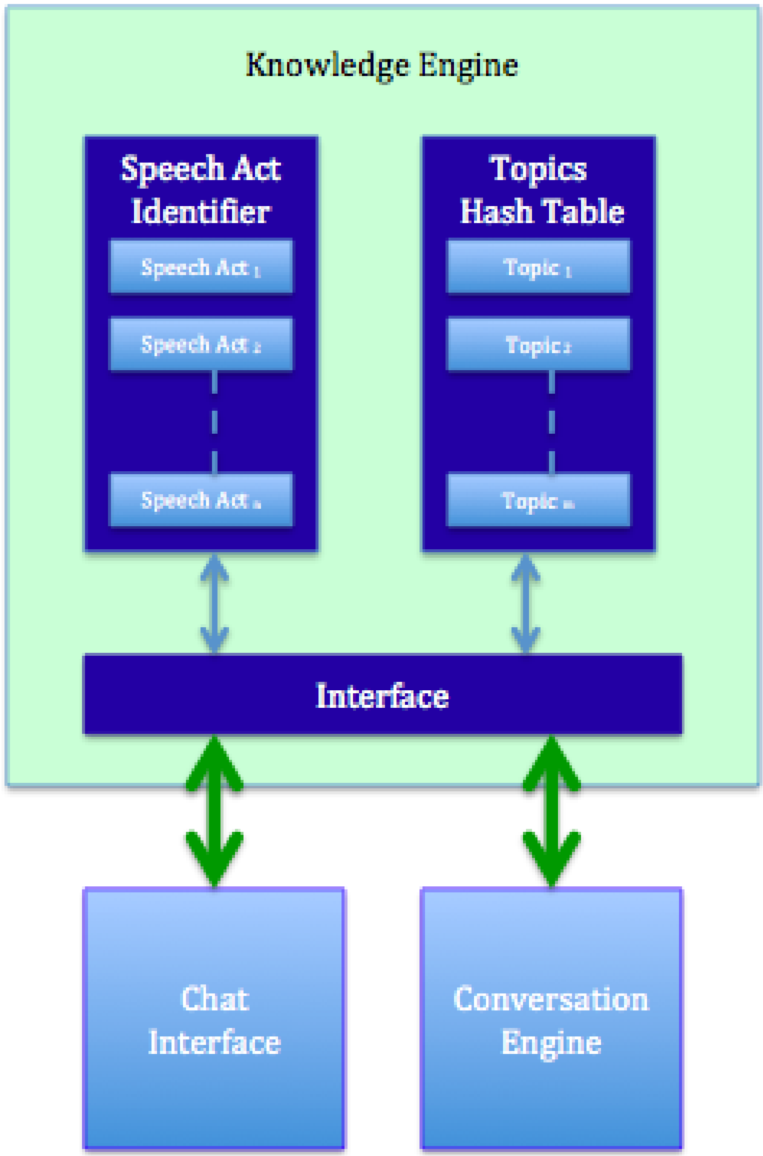

Figure 5.2: Knowledge Engine: Contains the Speech Act Identifier, and the Topics Hash Table. Each Topic consists of Context Maps. The blue block arrows show flow of information between the sub-modules within the Chat Interface. The green block arrows indicate flow of information between the modules external to it

1. **The Assertive Speech Act:** The utterance states a true fact about some state of the world pertaining to the context of the conversation and all involved participants commit to its veracity. This is an example of Searle's illocutionary

speech act [80, 79].

For example, the following are examples of Assertive Speech Acts:

- This restaurant makes the best burritos.
- John is the boss of this team.
- Your password is invalid.
- You have $2,000 in your account.

2. **The Directive Speech Act:** The utterance that serves as a request, command, advice, or instruction from one participant to another in terms of a specific action pertaining to the context of the conversation. This is an example of Searle's illocutionary speech act [80, 79].

   For example, the following are examples of directive speech acts:

   - Please bring me a breakfast burrito.
   - Stop making large trades when you have small margins.
   - Please close my account at the end of the month.
   - You should update your password immediately for security reasons.

3. **The Commissive Speech Act:** The utterance that serves as a commitment or promise to a future course of action that will change the state of the world pertaining to the context of the conversation by a participant, and all other participants commit to it's veracity. This is an example of Searle's illocutionary speech act [80, 79].

   For example, the following are examples of commissive speech acts:

   - I am going to eat burritos for dinner tonight.
   - I am going to update your account information.
   - You margin will increase when your trades are realized in 48 hours.

    * Your account will no longer be valid when your written request for change of status is accepted.

4. **The Expressive Speech Act:**The utterance expresses some emotion or attitude about the context of the conversetion. For example, the emotion could be gratitude, regret, excuse, etc. This is an example of Searle's illocutionary speech act [80, 79].

   For example, the following are examples of expressive speech acts:

    * Eating burritos at this restaurant makes me so happy.
    * I am sorry that you lost access to your account.
    * The process of canceling my trade requests is very cumbersome.
    * It is not fair that I have to pay a commission of USD 200 on a transaction that I did not explicitly authorize.

5. **The Declarative Speech Act:** The utterance causes the state of the world to change, with full implicit or explicit acknowledgement of the participants of the conversation .This is an example of Searle's illocutionary speech act [80, 79].

   For example, the following are examples of declarative speech acts:

    * This restaurant just lost my business.
    * John has been promoted to manager of the team.
    * Your account has been upgraded to premium status.
    * USD 3,000 has been credited to your account as your entire holdings in ACN have been liquidated.

In addition to the above 5 speech acts from Searle's illocutionary taxonomy [80, 79], I define the one more dialogue act, specific to customer service conversations.

6. **The Goal-Fulfillment Speech Act:** The utterance causes the state of the conversation to reach a conclusion, when all the issues raised in the conversation have been resolved and acknowledged by the participants of the conversation .This is an example of dialogue act specifically defined for this domain and context, i.e., resolving customer-service issues.

   For example, the following are examples of goal-fulfillment speech act::

   * That will be all for now, thank you.
   * Yes I am able to resolve the issue now.

Although only these 6 speech acts have been used in this implementation, the architecture is general enough to use all 42 speech acts from the modified SWBD-DAMSL tag set [38]. It can also use specialized dialogue acts, that are tailored to the domain for which the conversations will be generated. The speech acts are determined using a bag-of-words based latent semantic approach that is described in detail in chapter 6. The module for determining the bag-of-words is part of the Chat Interface, which will be described in detail later in this chapter.

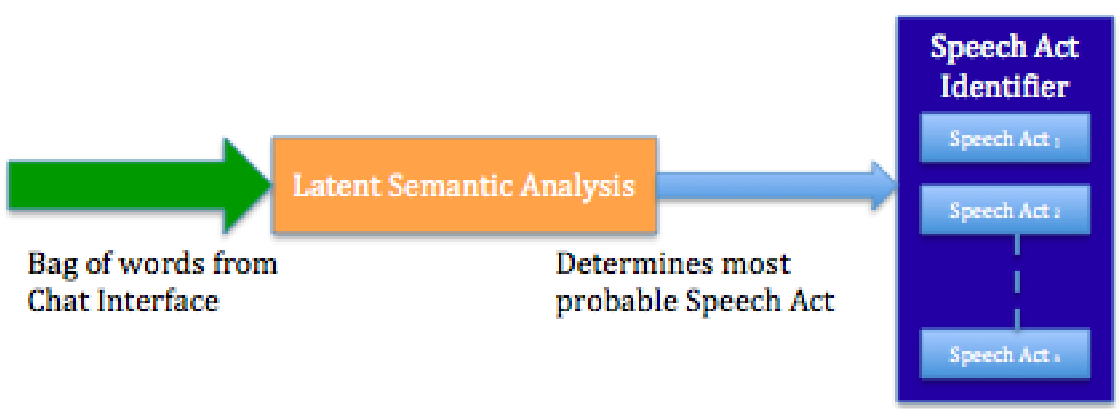

Figure 5.3: The bag of words captured by the Chat Interface is used by a Latent Semantic Analysis Algorithm to determine the most probable Speech Act associated with the utterance.

5.3.2 Topics Hash Table

As discussed in chapter 3, there are several methods to represent domain information for conversations. Chapter 3 described Ontologies and goal-fulfillment maps for storing large amounts of information in an easily scalable and maintainable manner. While Ontologies are formally comprehensive and robust feature-rich information repositories, for this work goal-fulfilment maps have been used to encode domain knowledge. As will be clear in subsequent discussions in this section, the purpose-driven nature of customer-service conversations lends themselves very nicely to have their background knowledge stored as a collection of goal-fulfillment maps. These data structures also adhere to the philosophy of the architecture being easy to update, maintain, and plug-and-play with various ideas.

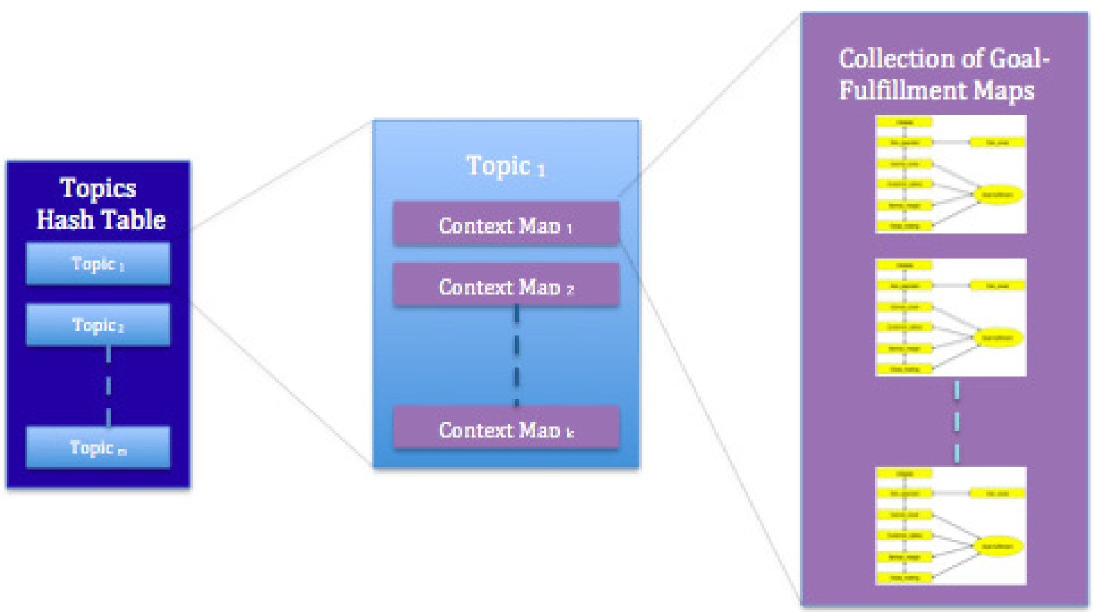

Figure 5.4: The Topics Hash Table encodes the background knowledge for the conversation. It is a hash map, where the values are individual topics. Each topic is implemented by a collection of contexts, where each context is a type of conversation. Each context is implemented as a collection of goal-fulfillment maps.

The Topic Hash Table organizes the set of topics in the scope of the conversation of the chatter bot. Each topic is a broad area of material for the particular domain implemented. Each specific topic entry in the hash table consists of a set of context maps that encode specific information about the topic. As described in the Knowledge Representation chapter, this can be a simple goal-fulfillment map, or a complex ontology. At present, I have simple goal-fulfillment graphs for a few topics. This can be easily updated using ontologies of higher complexity.

In this dissertation, conversation from nine different topics have been implemented, i.e., the background knowledge needed to support conversations pertaining nine topics in the domain have been encoded using the method described. The 9 topics can be classified under three main subject headings:

* **Accounts:** Conversations related to general account issues.
  - **Login:** The customer has problems logging in to the account.
  - **Configuration:** The customer wants to change or delete certain configurations in the account.
  - **Access:** The customer is unable to access certain parts of the account.
* **Balance:** Conversations related to issues regarding balances.
  - **Margins:** The customer perceives incorrect or inconsistent margin amounts in the account.
  - **Transfers:** The customer has problems or questions regarding transfer of balances between different portfolio instruments.
  - **Portfolio:** The customer wants to modify or give details of portfolio holdings.
* **Transactions:** Conversations related to transactional issues.

- **Commissions:** The customer has questions regarding commissions charged for completed or future transactions.
- **Orders:** The customer wants to modify pending orders or has questions regarding completed orders.
- **Processing:** The customer wants to resolve issues regarding processing of transactions or orders.

The topics are arranged in the form of a hash map. Only one topic is selected for every conversation, which is determined using a bag-of-words based latent semantic analysis approach.

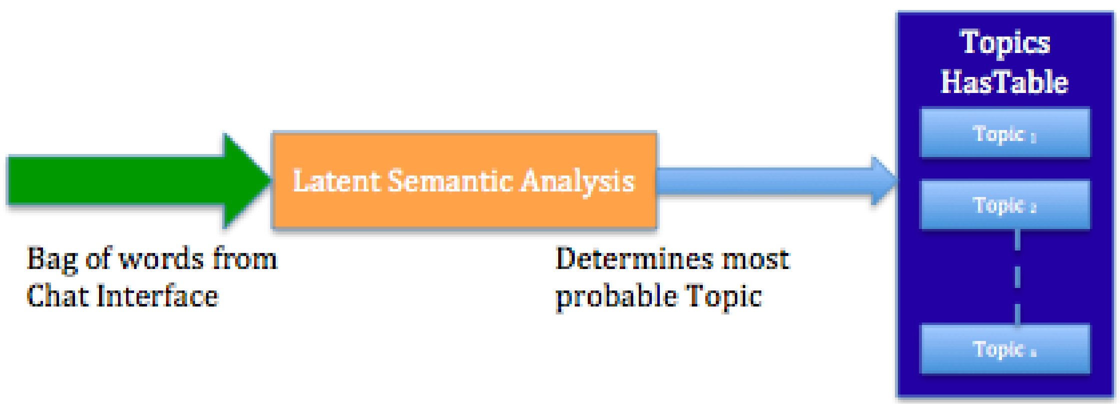

Figure 5.5: The bag of words captured by the Chat Interface is used by a Latent Semantic Analysis Algorithm to determine the most probable Topic associated with the utterance.

Within each topic element in the hash map, the information pertaining to the topic is arranged as a collection of context maps. The contexts are also arranged as hash maps. The contexts corresponding to the four main types of conversations that will be described in the Conversation Engine subsection and also some contexts common to all types of conversations.

* **Greeting:** This is common to all conversations. It models the small talk that

happens towards the beginning of each conversation.

* **Procedural:** This is a type of conversation described in the next subsection.

* **Toubleshooting:** This is a type of conversation described in the next subsection.

* **Dispute Resolution:** This is a type of conversation described in the next subsection.

* **Informational:** This is a type of conversation described in the next subsection.

* **Ending:** This is common to all conversations. It models the wrapping up that happens towards the end of each conversation.

Within each of these contexts, the conversation is organized as a collection of goal-fulfillment maps, i.e., a hash set. In this work, each of these goal-fulfillment maps have been manually coded.

For example, for the greeting, the goal-fulfillment map shown in Figure 5.7 is used. The greeting is part of the small talk sub-module that is used at the beginning of every conversation. This will be common to all topics. Now consider, Figure 5.8 and Figure 5.9. Both are contained in the Topic Hash Set of "Account". Within the "Account" Hash Set, Figure 5.8 shows a goal-fulfillment map that determines whether the account has enough balance to cover a transaction and is pulled up during conversations of the Troubleshooting type. Similarly, Figure 5.9 shows a goal-fulfillment map that encodes the knowledge to configure an account. This is pulled up during conversations of the type Procedural. Similarly, there are other goal-fulfillment maps in the Account Hash Set, depending on the situational context determined by the type of conversation. They are stored, indexed, and retrieved using a hash function. While such a hash function may not be necessary for this proof-of-concept system, it will be a necessary design element for a large system with many topics being endowed by hundreds of goal fulfillment maps across tens of situational contexts encoded by conversation types.

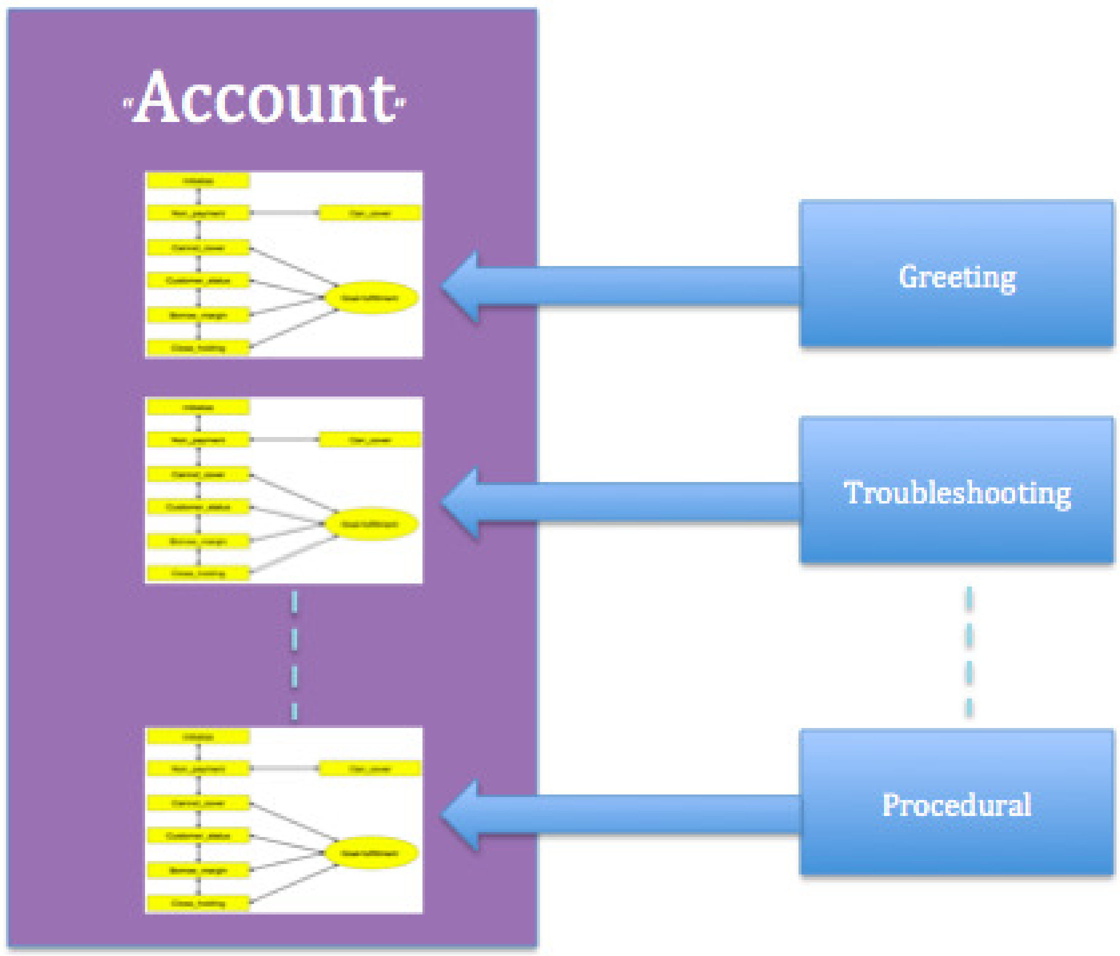

Figure 5.6: The context maps for the topic Accounts arranged in the form of a hash set.

5.4 Conversation Engine

The Conversation Engine models the semantics of the conversation. Along with content, semantics is the other component of a good conversation. An unique contri-

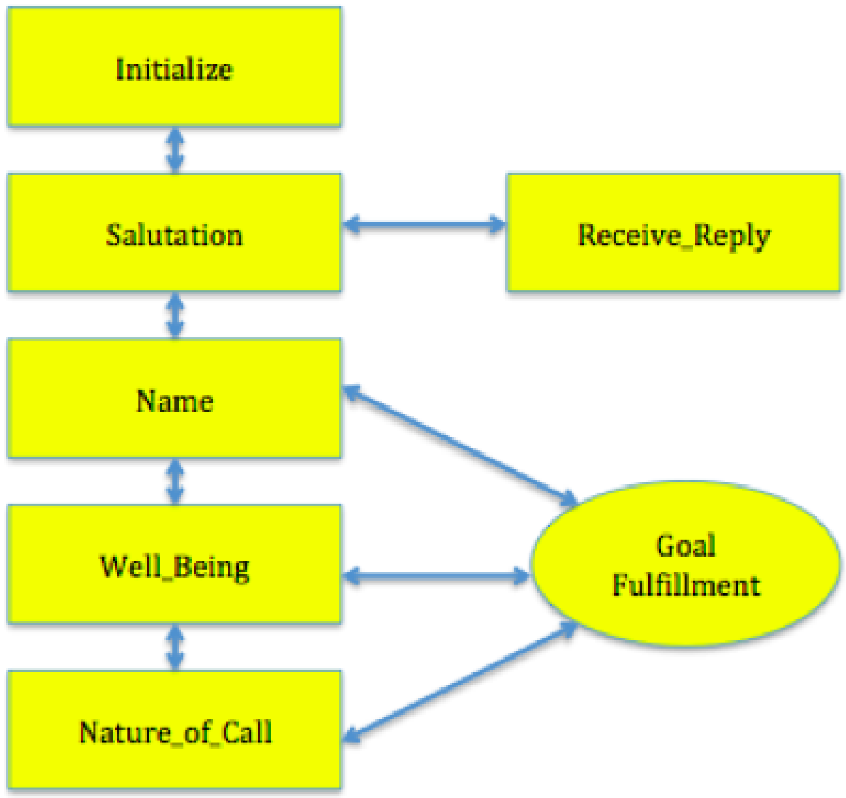

Figure 5.7: A goal-fulfillment map [58, 57, 60] that encodes the small talk that usually precedes every conversation.

bution of this research is that the modular nature of the architecture allows content to be distinctly modeled from semantics. While the previous section demonstrated how the content can be implemented using goal-fulfillment maps, this section demonstrates modeling the semantics of conversations in a modular fashion.

The Conversation Engine contains two modules, the Probabilistic Finite State Automaton, which stores the four different types of conversations described later, and the Conversation Planner, which decides the type of conversation to select, and keep track of the state of the conversation. The block diagram for the Conversation Engine is shown in Figure 15. The components of the Conversation Engine are described in detail below.

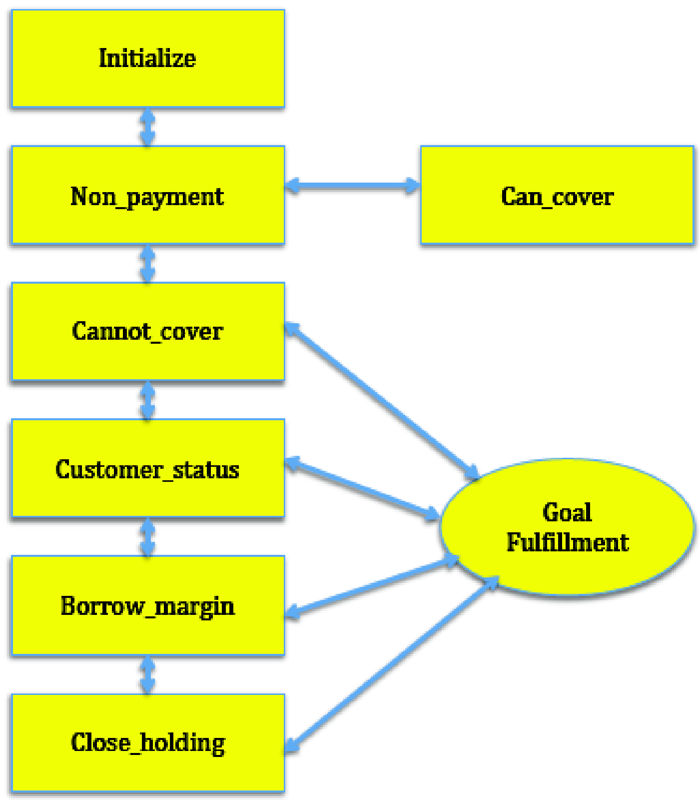

Figure 5.8: A goal-fulfillment map [58, 57, 60] that encodes that segment of the conversation which determines whether the margin in the customer's account is sufficient to cover the transaction requested.

5.4.1 Probabilistic Finite State Automaton

The conversational semantics have been implemented using several probabilistic finite state automata, where states represent semantic states of the conversation, transitions represent the speech act associated with the customer utterances, accepting states are the satisfaction and conclusion states, and non-accepting states are the dissatisfaction states. The conversations states have been manually identified from the corpus of human conversations in a customer service domain, i.e. transcripts of chat conversations between a human customer and a human customer service representative for an online electronic trading website. The transition probabilities

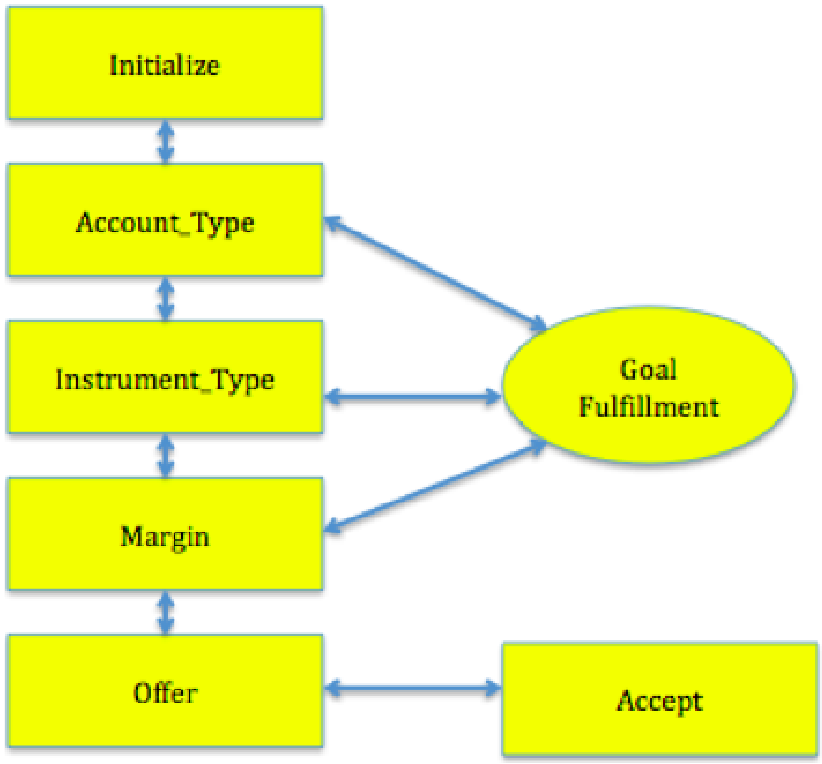

Figure 5.9: A goal-fulfillment map [58, 57, 60] that encodes the segment of conversation required to offer the most suitable account configuration to the customer.

are learned from the corpus. The stochastic automata are conversational grammars, which define the production rules for the particular type of conversation. This is distinct from regular language grammars that proceed individual sentences. In this research, the underlying mechanism to generate individual sentences is abstracted out, i.e., it is assumed to exist, and the conversational grammars are built on top of it. In this proof-of-concept work, the sentence producing mechanisms have not been implemented. But they would be needed in a real conversational system. The conversation control algorithm the Conversation Planner is responsible to maintaining track of the state of the conversation.

From analysis of the corpus, four distinct conversations have been identified.

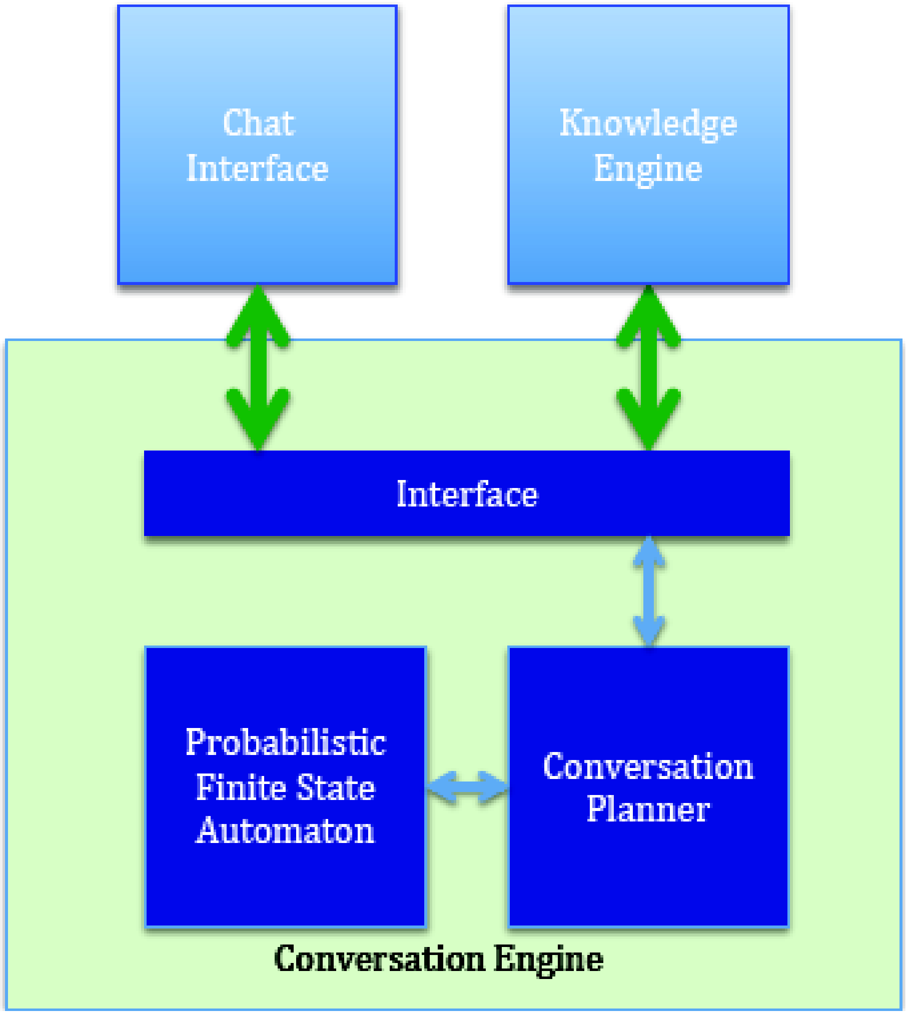

Figure 5.10: The Conversation Engine: Contains the Probabilistic Finite State Automaton and the Conversation Planner. The blue block arrows show flow of information between the sub-modules within the Chat Interface. The green block arrows indicate flow of information between the modules external to it

They are described in detail next.

Procedural Conversation A procedural conversation is one in which the chatter bot guides the customer though a series of steps to achieve some objective. For example, configuring an account, transferring balance from savings to trading account, etc. The stochastic finite state automata is shown in Figure 5.11. A procedural conversation consists of the following states.

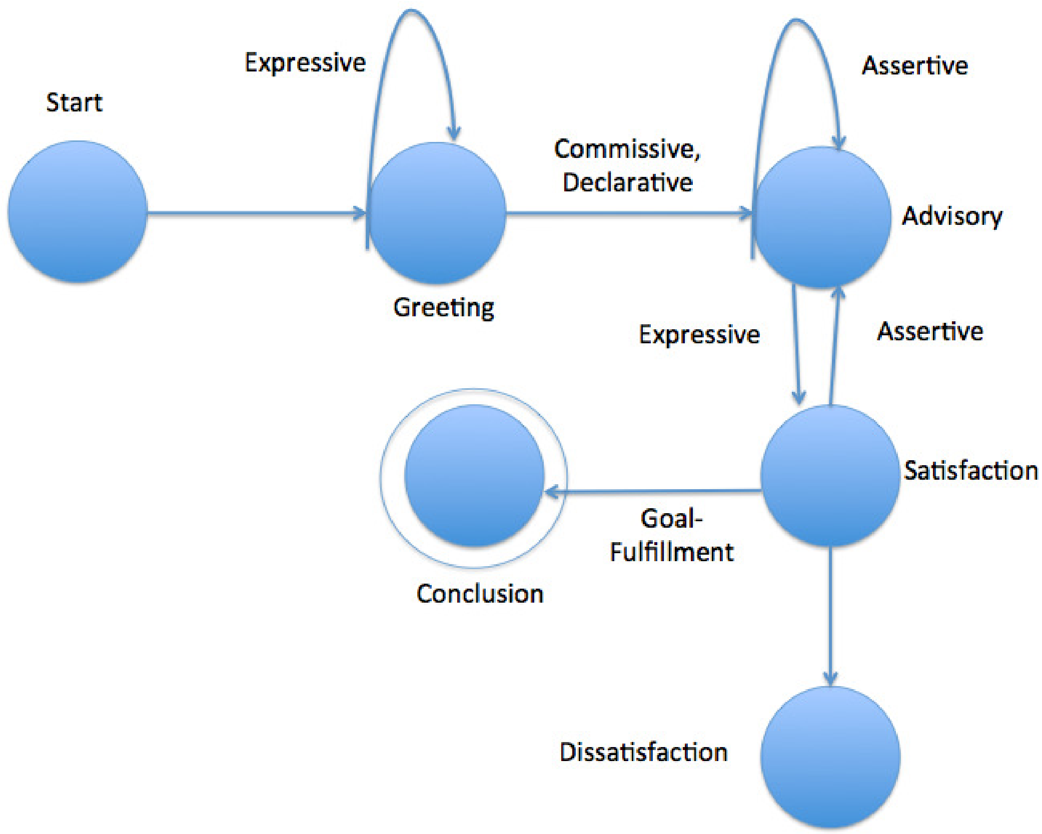

Figure 5.11: The Finite state automaton for Procedural Conversations.

* **Start:** The first state in which the conversation resides. This is the state before the conversation begins.

* **Greeting:** In this state, the conversation goes through the small talk process. Names and pleasantries are exchanged. The exact details of what has to be said implemented by the corresponding goal-fulfillment map in the Topic Hash Table.

* **Advisory:** In this state, the chatter bot instructs the customer though a series of steps to be followed or vice versa. Since this is essentially the customer or chatter bot asking each other to change the state of the world through actions, the utterances associated with the Assertive speech act keeps the conversation in the same state.

* **Satisfaction:** This state indicates that the procedure has been completed as verified by utterances associated with the Expressive speech act. An Assertive speech act can take the conversation back to the Advisory state.

* **Dissatisfaction:** This is a "dead-end" state that is reached when the conversation has reached beyond the programmatic limits of the chatter bot. Reaching this state indicates conversation failure, since it is not possible to leave this state.

* **Conclusion:** This state indicates the end of the conversation process. In algorithmic terms, reaching this state causes program control to be handed back to the Conversation Control algorithm. This state is reached when the last goal-fulfillment task as indicated by the corresponding Topic Hash table has been achieved.

Informational Conversation An informational conversation is one in which the chatter bot provides the customer with a series of facts or truths. For example, a list of all the pricing options for an account, all the special features associated with an account, the steps to be taken to conduct a transaction, etc. The stochastic finite state automata is shown in Figure 5.12. An informational conversation consists of the following states.

* **Start:** The first state in which the conversation resides. This is the state before the conversation begins.

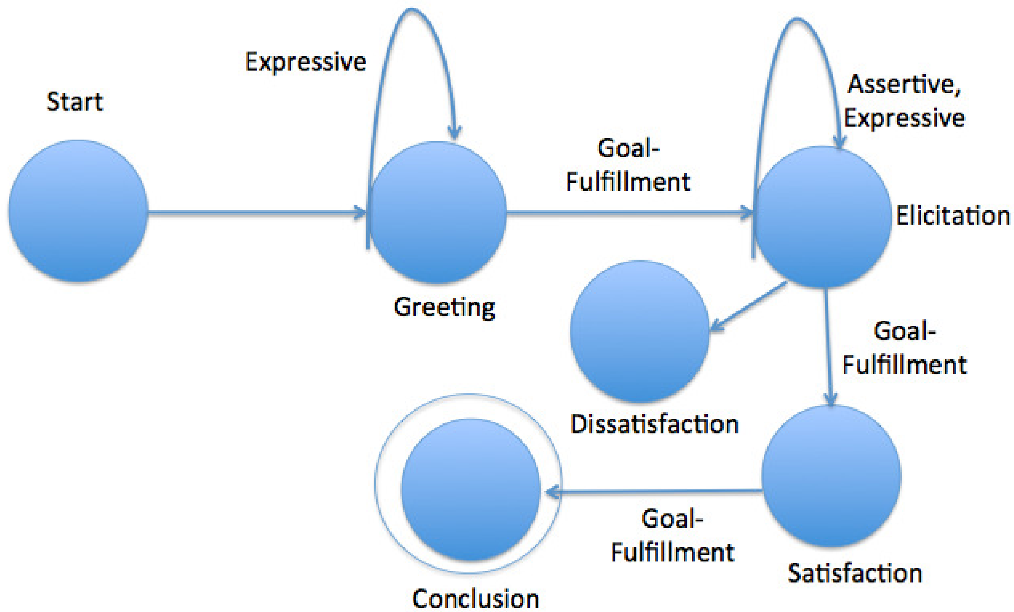

Figure 5.12: The Finite state automaton for Informational Conversations.

* **Greeting:** In this state, the conversation goes through the small talk process. Names and pleasantries are exchanged. The exact details of what has to be said implemented by the corresponding goal-fulfillment map in the Topic Hash Table.

* **Elicitation:** In this state, the chatter bot states truths or facts about the state of the world from the relevant situational context that is encoded by the correspond goal-fulfillment map in the Topic Hash Table. Utterances associated with both Assertive and Expressive speech acts keep the conversation in this state. This state is left only when Goal-Fulfillment is achieved as indicated by the corresponding map in the Topic hash Table.

* **Satisfaction:** This state indicates that the procedure has been completed as verified by utterances associated with the Expressive speech act. An Assertive speech act can take the conversation back to the Advisory state.

* **Dissatisfaction:** This is a "dead-end" state that is reached when the conversation has reached beyond the programmatic limits of the chatter bot. Reaching this state indicates conversation failure, since it is not possible to leave this state.

* **Conclusion:** This state indicates the end of the conversation process. In algorithmic terms, reaching this state causes program control to be handed back to the Conversation Control algorithm. This state is reached when the last goal-fulfillment task as indicated by the corresponding Topic Hash table has been achieved.

Notice that there is a subtle difference between a procedural and informational conversation. In procedural conversations, utterances belong to Expressive speech act cause the conversation to leave the Advisory state. In informational conversations, utterances belonging to the Expressive speech act causes the conversation to remain in the Elicitation state. This is an important consideration for the underlying conversation semantics since this means the two types of conversations are generated by different conversational grammars.

Troubleshooting Conversation A troubleshooting conversation in which the customer has a problem, and the chatter bot takes steps to understand the nature of the problem, and then guides the customer through a series of steps to either find more information about the problem or try to resolve it. For example, troubleshooting an incorrect balance showing in the trading account, trouble an issue with incorrectly executed trades and transactions, etc. The stochastic finite state automaton is shown in Figure 5.13. A troubleshooting conversation consists of the following states.

* **Start:** The first state in which the conversation resides. This is the state

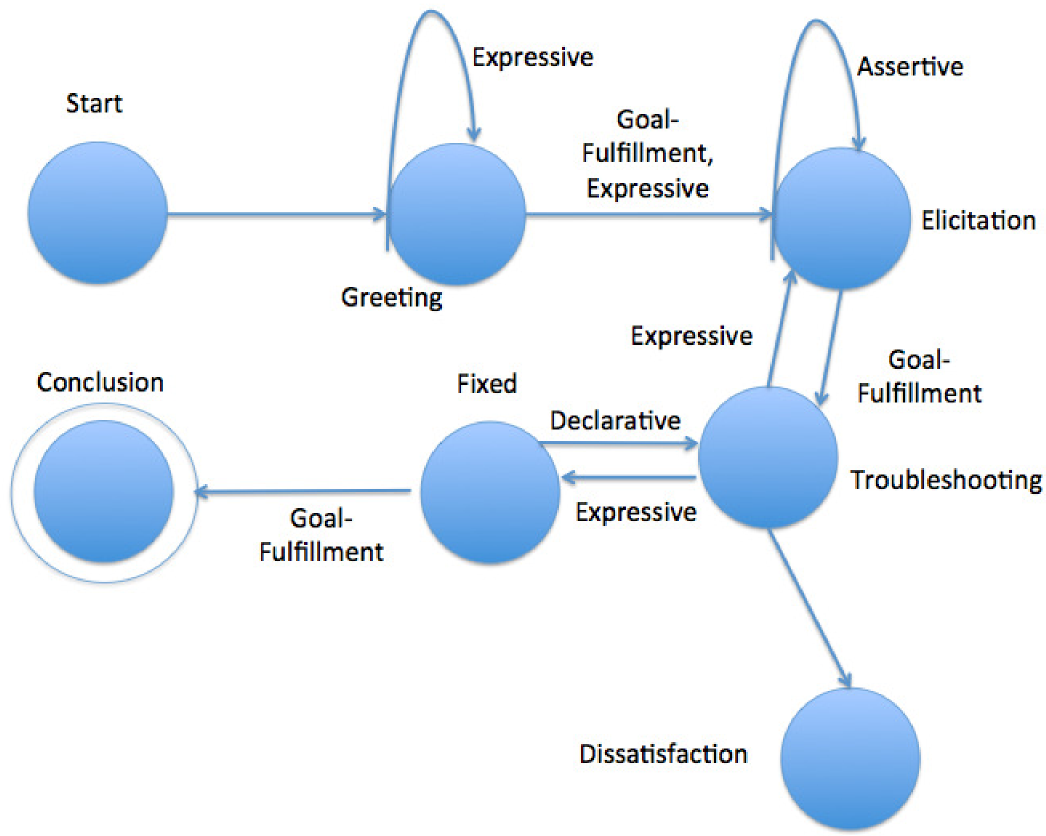

Figure 5.13: The Finite state automaton for Troubleshooting Conversations.

before the conversation begins.

* **Greeting:** In this state, the conversation goes through the small talk process. Names and pleasantries are exchanged. The exact details of what has to be said implemented by the corresponding goal-fulfillment map in the Topic Hash Table.

* **Elicitation:** In this state, the chatter bot asks questions, and solicits information about the state of the world from the relevant situational context that is encoded by the correspond goal-fulfillment map in the Topic Hash Table. Utterances associated with Assertive speech acts keep the conversation in this state. This state is left only when Goal-Fulfillment is achieved as indicated by

the corresponding map in the Topic hash Table. Note that while an utterance belonging to an Expressive speech act would keep the conversation in this state, such an utterance is unlikely according to the probabilities learned from the corpus. Encountering such an utterance would mean that the Conversation has incorrectly identified the conversation to be of a troubleshooting type.

* **Troubleshooting:** In this state, the chatter bot makes the customer take proactive steps to change the state of the world in an attempt to resolve the problem issue. The set of steps is once again defined by the corresponding goal-fulfillment map in the Topic Hash Table. Note that alternative utterances belong to the Declarative and Expressive speech acts would take the conversation back and forth between the Troubleshooting and Fixed state. Any other speech act utterance will likely take the conversation to the Dissatisfaction state.

* **Fixed:** This state indicates that the problem issues has been partially or completely fixed. A partial fix would trigger an Expressive speech act utterance from the customer which would take the conversation back to the Troubleshooting state. Only a Goal-fulfillment speech act, as indicated by the goal-fulfillment map in the corresponding Topic hash table would take the conversation to the conclusion state.

* **Dissatisfaction:** This is a "dead-end" state that is reached when the conversation has reached beyond the programmatic limits of the chatter bot. Reaching this state indicates conversation failure, since it is not possible to leave this state.

* **Conclusion:** This state indicates the end of the conversation process. In algorithmic terms, reaching this state causes program control to be handed back to the Conversation Control algorithm. This state is reached when the

last goal-fulfillment task, as indicated by the corresponding Topic Hash table, has been achieved.

Dispute Resolution Conversation A dispute resolution conversation is one in which there is customer disagrees with something, and the chatter bot tries to resolve the disagreement. For example, the customer feels than an incorrect commissions fee has been charged due to misidentification of transaction parameters, the customer feels that the transactions were incorrectly executed due to misinterpretation of instructions, etc. The stochastic finite state automata is shown in Figure 5.14. A dispute resolution conversation consists of the following states.

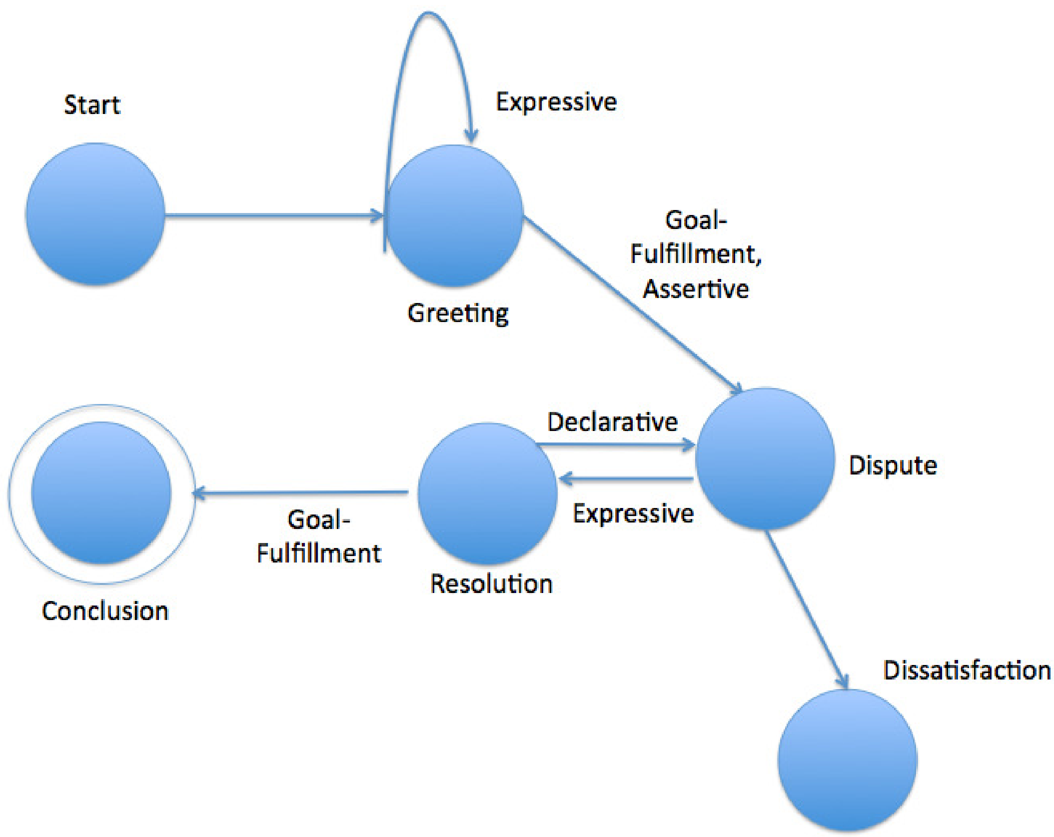

Figure 5.14: The Finite state automaton for Dispute Resolution conversations.

* **Start:** The first state in which the conversation resides. This is the state before the conversation begins.

* **Greeting:** In this state, the conversation goes through the small talk process. Names and pleasantries are exchanged. The exact details of what has to be said implemented by the corresponding goal-fulfillment map in the Topic Hash Table.

* **Dispute:** In this state, the chatter bot takes steps to resolve the dispute by asking the customer to take proactive steps to change some truth about the state of the world, or empathize or criticize some aspect of the customer situation or statement. These courses of action are again guided by the corresponding goal-fulfillment map in the Topic Hash table. An assertive statement is likely to send the conversation to the Dissatisfaction state, since it will indicate escalation of the dispute, possible irrevocably.

* **Resolution:** This state indicates that the problem issue has been partially or completely fixed. A partial fix would trigger a Declarative speech act utterance from the customer which would take the conversation back to the Dispute state. Only a Goal-fulfillment speech act, as indicated by the goal-fulfillment map in the corresponding Topic hash table, would take the conversation to the conclusion state.

* **Dissatisfaction:** This is a "dead-end" state that is reached when the conversation has reached beyond the programmatic limits of the chatter bot. Reaching this state indicates conversation failure, since it is not possible to leave this state.

* **Conclusion:** This state indicates the end of the conversation process. In algorithmic terms, reaching this state causes program control to be handed back to the Conversation Control algorithm. This state is reached when the

last goal-fulfillment task as indicated by the corresponding Topic Hash table has been achieved.

A dispute resolution conversation differs from a troubleshooting conversation because the former doesn't have a corresponding Elicitation state. Hence, a conversation resembling an argument is handled differently compared to a conversation seeking to solve problems. In practice, it turns out that the troubleshooting and dispute resolution conversations are harder to model and hence are more interesting to study.

5.4.2 Conversation Planner

The conversation planner serves as the workspace for the conversation generation as shown in Figure 5.15. This module decides which type of transition to perform in the stochastic automaton, maintains the likeliness score for each type of conversations, and maintains a counter to keep track of the state for the conversation. These functions are described below.

1. **Maintain workspace for the conversation:** The conversation planner maintains four simultaneous solutions corresponding to the four conversations types. With every utterance in the conversation, each conversation solution is updated. When one solution becomes highly likely as indicated by a heuristic score described later, then that solution is maintained and all others removed from the workspace. Conversely, when one solution becomes highly unlikely according the heuristic score described below, it is removed from the workspace. There have been occasions when all the solutions have had their heuristic scores fall below the pre-defined threshold and dropped from the workspace. This leads to conversation failure.

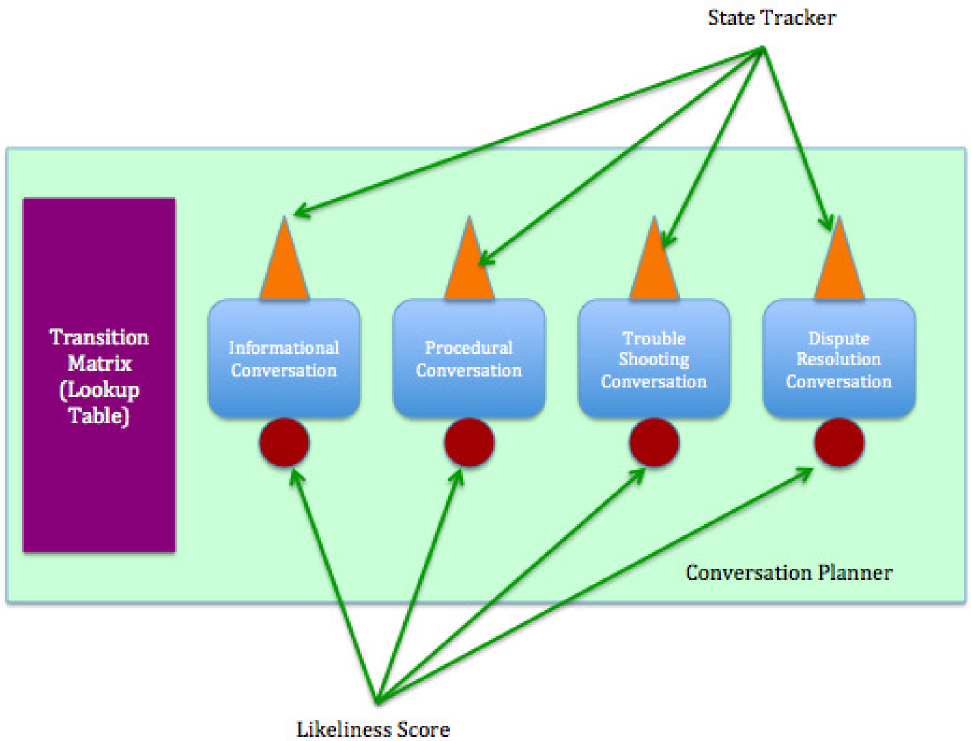

Figure 5.15: The Conversation Planner consisting of the transition matrix lookup table, the state tracker, the likeliness score variable, and the conversation solutions in the workspace.

2. **Transition matrix probability values:** The conversation planner has access to a lookup table of transition probabilities for the four stochastic automata corresponding to the four types of conversation solutions. These probability values are learned from the corpus and is described in the next chapter. This lookup table indicates which transition to make for each automata.

3. **State tracker:** For each of the conversation solutions still active in the workspace, a separate tracker is maintained to indicate the current state of the conversation according to that solution. There can be from zero (conversation failure) to four state trackers present simultaneously.

4. **Likeliness Score:** For each of the four solutions, a likeliness score is main-

tained. This score is an indication of how likely it is that a solution is the right one for the specific conversation being generated. This score is an integer variable initialized to 0. When the conversation follows the most likely path as indicated by the automata corresponding to the variable, it is incremented by 3. Since initially, all conversations have the "Greeting" stage that involves small talk, each likeliness score variable is increased by 3 for every progression of conversation state. When the next progression does not agree with the one indicated by the automaton, the score is decreased by 1. Scores that fall below 0 are dropped from the workspace, since that indicates that the conversation has drifted sufficiently from the path suggested by the corresponding automaton and hence that solution is unlikely.

When one score becomes an $n-th$ multiple of the next highest score, for $n > 8$, where n indicates the number of turns of utterances, then all other solutions are dropped front the workspace. This situation is an indication that one of the solutions is overwhelmingly likely as compared to the others and should be the only one considered. As of now, there is not a very principled way to determine the value or n, or the heuristic values 3 and 0. It has been selected purely through trial and error, and repeated tweaking. Such a solution was implemented because there is some evidence that human being process conversation in this way [9, 13]. Literature from linguistic neurobiology suggests that human beings maintain several alternative solutions while processing conversations [11, 102].

5.5 Chat Interface

The Chat Interface is the module that directly interfaces with the user. Its high-level function is to receive chat text from the user, pre-process this text and pass it on to the Knowledge Engine and the Conversation Engine, receive input back from the engines, and then transmit chat text back to the user. It has several sub-modules that facilitate this task. The block diagram is described in Figure 5.16.

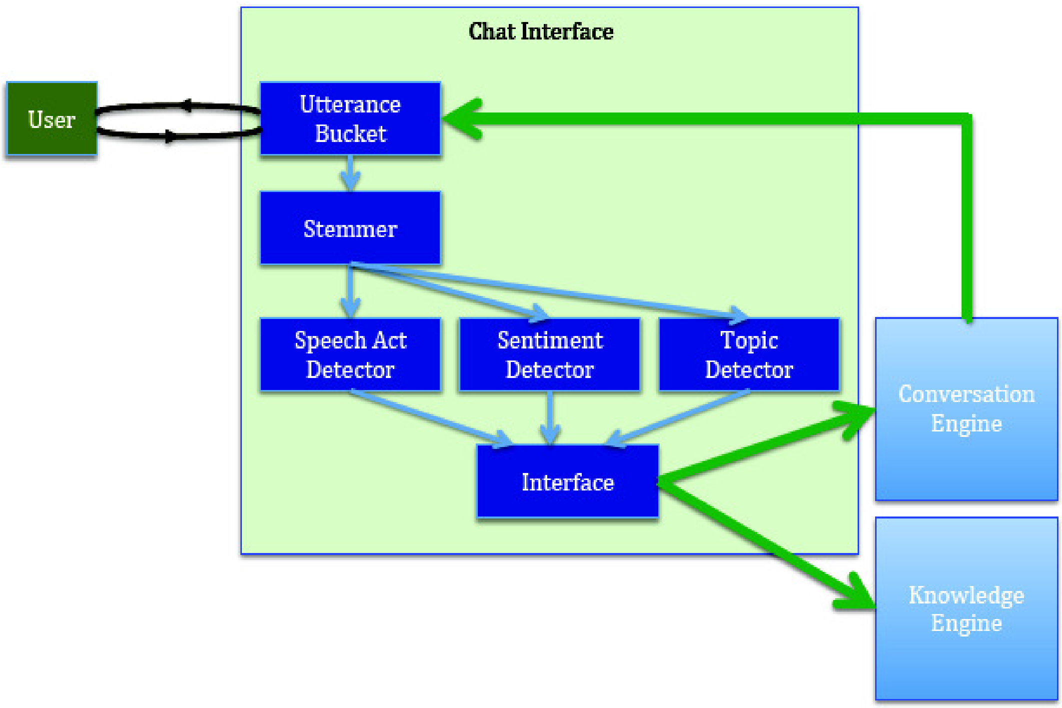

Figure 5.16: The Chat Interface: This module directly interfaces with the user. The blue block arrows show flow of information between the sub-modules within the Chat Interface. The green block arrows indicate flow of information between the modules external to it

5.5.1 Utterance Bucket

The Utterance Bucket is an interface that receives the chat text from the user, and places the text in a buffer. This module is designed to have a library function to correct spellings and grammar. But it is currently left unimplemented.

5.5.2 Stemmer

The Stemmer module reduces text to its root stems. For example, all the inflected words *troubles*, *troubled*, and *troublesome* are reduced to the root form *trouble*. This is implemented using an open-source version of the classic Porter Stemming Algorithm [71]. Then the stemmed set of words is subjected to the NLTK parser that is part of the open source NLTK suite. This extracts a set of discerning keywords.

5.5.3 Speech Act Detector

This module examines the keywords and detects the Speech Act associated with the keywords. The module has access to a set of keywords stored in a hash set. It detects which speech act keywords are present in the text, and puts them in a collection.

5.5.4 Sentiment Detector

This module detects sentiment associated with the utterance. The standard set of bag of words pertaining to sentiments is used [64]. The set of sentiments pertaining to customer service issues, which is stored in a searchable hash table. Currently, this module is left unimplemented.

5.5.5 Topic Detector

This module detects the topic for the conversation. Like the Speech Act detector module, the topic module simply detects which topic keywords are present by referring to a hash set of topic keywords and places these words into a collection.

5.5.6 Interface

This module collates the information from the Speech Act, Sentiment, and Topic detectors, in to a single stream of arguments, and then passes it on to the Conversation Engine, and the Knowledge Engine.

5.5.7 Anatomy of a Conversation

This section shows how a conversation is created in a step-by-step fashion through the architecture.

1. The conversation starts with a human making a comment.

   Customer : `I would like to open a new account for day trading.`
   `What are my options?`

   This message is entered from the standard terminal. The Utterance Bucket directly collects the text in the form of a string. A standard spellchecker and grammar checker autocorrects the spelling and grammatical errors in the sentence if any

2. The correct sentence, free of spelling and grammatical error, is sent to the Stemmer. Using Porter's Stemming algorithm, the following stems are obtained, "account", "day trade", "open", and "options".

3. The entire stemmed sentence is then passed on simultaneously to the Speech Act Detector, the Sentiment detector, and the Topic Detector. The following events then take place.

   * The Speech Act Detector uses Latent Semantic Analysis to determine that

the type of speech act is ”Expressive”, since he bag of words included ”would” and ”like”.

* The Sentiment Detector detects that the sentiment is neutral, since none of the words from the positive or negative bag of words is encountered.
* The Topic Detector determines using Latent Semantic analysis that the topic is ”new account” using bag of words ”new”, ”account”, and ”open”.

4. The output of the Speech Act Detector, the Sentiment Detector, and the Topic Detector is then sent to the interface. The Interface combines these into an array list, and sends the array list to the Conversation Engine and the Knowledge Engine simultaneously.

5. In the Knowledge Engine, the following steps take place.

   * The Interface of the knowledge engine receives the array list and sends it to the Speech Act Identifier. This module selects the correct speech act from the list as ”expressive”.
   * The interface also sends the bag of words to the topic hash table. The hash table retrieves the topic as ”new account”. The appropriate context map is then pulled out. This context map lists the steps for the encoded knowledge for opening a new account in the form of a goal-fulfillment map. The appropriate goal-fulfillment map, shown in Figure 5.17, is then put in to the workspace and sent to the interface.
   * A goal-fulfillment algorithm is initiated. A counter is initiated to keep track of the progression of goals in the map.

6. In the Conversation Engine, the following steps take place.

   * In the Probabilistic finite State Automata, initially all four possible solutions are maintained. This is because initially the probabilities of each

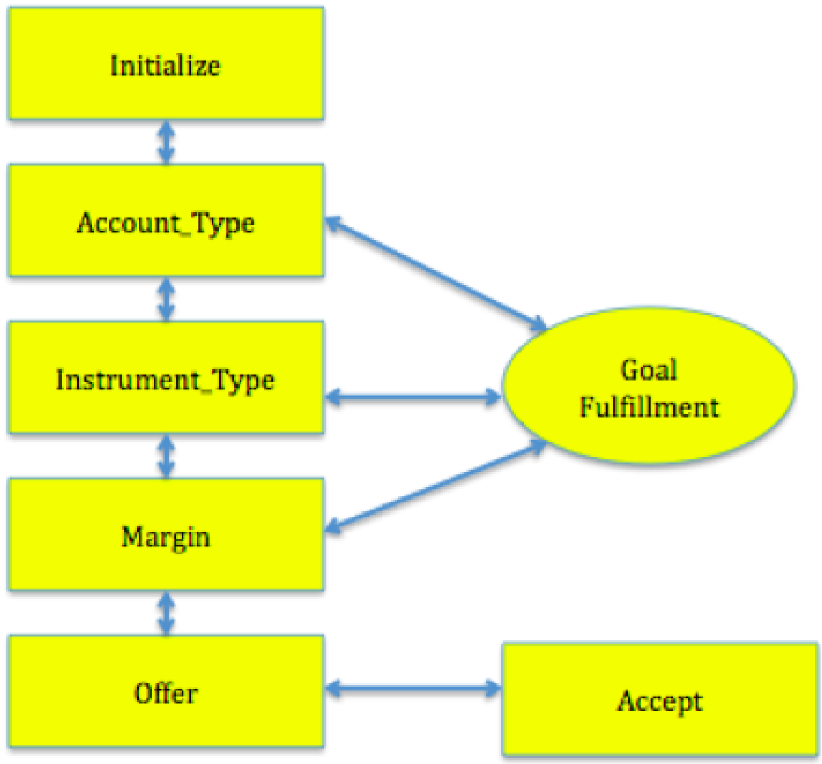

Figure 5.17: Goal-fulfillment map [58, 57, 60] selected by the Knowledge Engine in the anatomy of a conversation.

conversation type will be nearly equal. A counter is initialized to maintain the current state of the conversation in each solution.

* The Conversation Planner will calculate the probabilities of transition from one state to another depending upon the Speech Act being uttered. These transitions are learned from the corpus and are stored in a lookup table. The Conversation Planner is responsible for advancing the counter indicating the current state of the conversation.

7. The information is sent back to the Chat Interface. The Utterance Bucket corrects spelling (unlikely) and grammatical errors, and then outputs the response of the chatter bot to the standard terminal.

`Chatter Bot : Do you have an existing trading account or would you like to open a new one?`

8. This process is repeated until the end of the conversation is indicated by the Conversation Planner counter being in an accepting state.

5.6 Uniqueness of Architecture

The review of the literature in Chapter 2 revealed some common themes.

* All three main approaches: syntactic, stochastic and semantic, perform their analysis on a finer granular level, where each unit of data is a sentence. Topic modeling and other learning approaches look at a chunk of text as whole, without consideration of the transitory nature of the content.

* The literature doesn't have a standardized approach to knowledge representation for a conversation. Each domain has a specific architecture for representing knowledge appropriate for that domain. This precludes easy translation of techniques to other domains and scales poorly.

* No study in the literature represents conversations as a process that transitions across a set of states.

* Semantic approaches like sentiment analysis consider the presence or absence of certain words to determine polarity of a sentence. It is a static approach. It also fails take into consideration that a conversation is a process that transitions through several states.

The uniqueness of this work is that I demonstrate a modular, robust, and scalable architecture for chatter bots [6, 7, 8]. The specific concepts of pragmatics, speech acts, and dialogue acts are well known in the field of conversation theory [80, 79, 38, 30, 31, 32, 77, 26]. However, this dissertation is the first example of computationally modeling these specific concepts to realize pragmatic semantics for chatter bots [6, 7, 8]. Similarly, specific concepts like goal-fulfillment maps [59, 56, 58, 57, 60] have been explored previously in the knowledge representation literature. But this dissertation is the first example of using goal-fulfillment maps for modeling content semantics for a chatter bots in the form of a series of sub-contexts [6, 7, 8]. In addition, this dissertation is the first example of combining pragmatic semantics and content semantics to generate artificial conversations [6, 7, 8].

Chapter 6

Generation of Artificial Conversations

This chapter describes the steps taken to generate artificial conversations using the chatter bot architecture. The effort for generating artificial conversations was supported by a corpus of human to human conversations. The language parameters required to model content semantics and pragmatic semantics were learned from that corpus. Then the models were used to generate artificial conversations. Each of these steps are described in the next three sections.

6.1 Corpus

Since modeling spoken conversation is inherently different from modeling typed chat conversations, it was necessary to obtain a corpus of typed human to human chat transcripts. Also, since the domain being modeled was customer service situations where a human customer communicates with a chatter bot, it was necessary to have a corpus from this domain.

6.1.1 Chat Transcripts

The corpus that was used by this research consists of chat transcripts taken from logged chat sessions of an online electronic trading website called eTrade. The chat transcripts were of a human customer of eTrade communicating with a human customer service agent working for eTrade. The corpus was provided by Next IT Inc. of Spokane, WA. Next IT Inc. is an organization that designs chatter bots for deployment on client websites. eTrade was one such client, and agreed to release their corpus for research purposes. The author of this dissertation or the dissertation advisor should be contacted for access to the corpus.

Following were the characteristics of the corpus.

* The corpus consisted of 2,886 distinct conversations. Each conversation was in the form of an Excel file and was clearly demarcated by a unique conversation identifier.

* In each conversation, the utterances were marked by who was delivering it, either the customer or the customer service agent.

* The shortest conversation had only 5 distinct utterances. The longest conversation had 82 distinct utterances. The median was 26 utterances and the average was around 22 utterances.

* The utterances were mostly interleaved, i.e., alternating between the customer and the customer service agent. There were a few conversations that were not interleaved. But this wasn't considered sufficiently significant to change the analysis.

* Most of the conversations were related to single context. A few conversations were related to more than one context and these were not analyzed.

This corpus was used to learn the parameters of the language models mentioned in chapter 5.

6.1.2 Pre-processing of the corpus

The following pre-processing steps were carried out on the corpus, before it was mined. The corpus was extensively studied through manual scanning and simple text mining tools like counting words, nouns and verbs. This was aided by knowledge about the domain, i.e., customer service chat logs for an online electronic trading website. The following items were identified as a consequence of this analysis.

* The conversations in the corpus could be classified into four types: Procedural, Informational, Troubleshooting, and Dispute Resolution. While these types were not exhaustive, most of the conversation in the corpus belonged to one of these categories. Some conversations were ambiguous and could have been classified in more than one of these conversation types. Some conversations were clearly could not be classified in any of these main types. These conversations were ignored. These four types of conversations were specifically selected because distinct automata could be defined for them. These selections were done manually.

* Again by studying the corpus, several topics were selected: login, configuration, access, margins, transfers, portfolio, commissions, orders, and processing. For each of these topics, specific contexts were manually identified as follows:

  - Steps to recover a forgotten password. (Procedural, Troubleshooting)
  - Steps to deactivate an account. (Procedural, Troubleshooting)
  - Steps to reactivate a closed account. (Procedural, Troubleshooting, Dispute resolution)

- Steps to configure a new account for daily margin trading mode .(Procedural, Troubleshooting)
- Steps to configure a new account for regular margin trading mode. (Procedural, Troubleshooting)
- Steps to change instrument configuration for an existing account. (Procedural)
- Steps to access list of past transactions that were already executed. (Procedural, Troubleshooting, Dispute resolution)
- Steps to Increase the trading margin in the account. (Procedural, Troubleshooting, Dispute resolution)
- Conditions for account to show a lower trading margin than expected. (Troubleshooting, Dispute resolution)
- Conditions for fund transfers to show up in trading margin. (Troubleshooting, Dispute resolution)
- Steps for adding options to a portfolio. (Procedural, Troubleshooting)
- Steps for removing options from a portfolio. (Procedural)
- Rules for determining how much commissions should be charged for a transaction. (Informational, Troubleshooting, Dispute resolution)
- Conditions under which a higher commission can be charged. (Troubleshooting, Dispute resolution)
- Rules for placing orders. (Dispute resolution)
- Conditions for execution of orders already placed. (Troubleshooting, Dispute resolution)
- Steps for canceling orders already placed. (Dispute resolution)

  - Rules for the margin to reflect the results of sell orders, i.e., how long it takes for the amount to be added to the margin after the sell order has been processed. (Informational, Troubleshooting, Dispute resolution)
  - Rules for maintaining sufficient margin to execute buy orders. (Troubleshooting, Dispute resolution)
  - Steps to verify the details of the orders like number of units, date, or total amount. (Informational, Troubleshooting, Dispute resolution)

* By referring to the actual human conversations in these contexts, the specific domain knowledge was for the above contexts was manually obtained, i.e., the access protocols for logins, the specific number of days required to update margin after transaction, the specific dollar amount for commissions charged, etc.

* Using the information obtained in the previous steps, the goal-fulfillment maps for all the above combination of contexts and conversation types were manually created and encoded in the Topic Hash Table.

6.2 Parameters of a Conversation

The corpus was then used to learn the conversation parameters, i.e., the speech acts and the transition probabilities.

6.2.1 Learning Transition Probabilities

* For each of the four conversation types, Procedural, Information, Troubleshooting, and Dispute Resolution, fifty conversations were selected from the corpus for a total of 200 sample conversations. From these conversations, the states

of the conversation as defined by the finite state automata were annotated manually.

* The transitions between each pair of states in the automata were then counted. The speech act associated with the utterance that caused the transition was also noted, as described in section 6.6.2.

* Using this information, the transition probability matrix for all four types of conversations was calculated. This was preserved as a lookup table in the Conversation Planner of the Conversation Engine.

* The conversations were generated using the probability values in these automata. See Figures 5.11, 5.12, 5.13, and 5.14 for the automata. In these figures, each of the states, in theory, have six possible transition for each of the six pre-defined Speech Acts. In theory, the transitions corresponding to the Speech Acts other than the ones shown in the figure have a probability of 0. In practice, they have a small probability value very close to zero. While generating a conversation, occasionally these low probability transitions would occur. On such occasions, either that utterance step was regenerated, or the transition matrix was manually tweaked to make the low probability transition value exactly 0, and a corresponding change was made to the probability values for the other transitions to preserve the rules of probability, i.e., make sure the sum of the probability of all possible transitions is unity.

6.2.2 Learning Speech Acts

* For each of the Speech Acts, Assertive, Directive, Commissive, Expressive, and Declarative, the standard bag of words list as defined in lexical taxonomy of speech and dialogue acts [54] was used. In addition, this was supplemented by more words as follows:

- Assertives {"good", "bad", "incorrect", "lower", "higher", "best", "worst", "invalid", "correct", "incorrect", "wrong", "right", "unable", "able"}

- Directives {"close account", "change mode", "configure account", "change margin", "change portfolio", "change option"}

- Commissives {"will cancel account", "will reactivate account", "will deactivate account", "orders will be reinstated", "margin will be restored", "commissions will be removed"}

- Expressives {"glad", "happy", "upset", "unhappy", "unacceptable, "acceptable" }

- Declaratives {"account is closed", , "account is reactivated", "account is deactivated", "configuration is changed", "orders are cancelled", "orders are reinstated", "margin is restored"}

* Using these bags of words, each utterance in the sample of 200 conversations selected in section 6.2.1 was tagged with the appropriate speed act using latent semantic analysis.

* For the goal-fulfillment speech act, the bag of words {"Thanks", "Thank You", "Resolved", "Nothing Else", "That's All", "I am Good" } was used to tag the utterances. This was supplemented by some ad-hoc techniques as well. The last utterances of every conversation in the entire corpus (not just the sample of 200 conversations selected in the previous step) was tagged with the goal-fulfillment speech act, and those words were added to the bag of words.

* Goal-fulfillment maps were created by hand to encode this domain knowledge. This formed the base of the Knowledge Engine. This resulted in additional phrases { "Have a great day", "Have a good day", "You are welcome"}

* These bags of words were used to identify the speech acts in the Knowledge Engine in the previous chapter.

6.2.3 Learning Topic Acts

* From the sample of 200 selected conversations described in 6.2.1, for each of the topics, i.e., login, configuration, access, margins, transfers, portfolio, commissions, orders, and processing, the following bags of words were identified manually.

  - Login: { "Login", "Password"}
  - Configuration: {"Configuration", "Upgrade", "Daily", "Regular" }
  - Access: {"Access" }
  - Margins: {"Margins", "Balance" }
  - Transfers: {"Transfers", "Allocation" }
  - Portfolio: {"Portfolio", "Commodity", "Equity", "Trade" }
  - Commissions: {"Commissions", "Charge", "Cost" }
  - Orders: {"Orders", "Buy", "Sell" }
  - Processing: {"Processing", "Reinstate", "Cancel", "Execute" }

* This bag of words was used to define the topic model in the Knowledge Engine. These bags of words were used as keywords to identify the correct topics in the Topic Hash Table in the Knowledge Engine.

* On occasion, the bags of words were tweaked by hand in an ad-hoc fashion to adjust the selection of topics if certain trends were noticed.

6.3 Conversation Creation

The next step is to actually generate the artificial conversation using the chatter bot architecture. The conversations are generated by me, by interacting with the chatter bot architecture via a standard terminal. These are the steps to generate a conversation.

1. Play the role of the customer of the online electronic trading website. Pick out an issue from the list in 6.1.2. "Know" the responses to all the customer-side details. For example, know that the account can have two different modes and two different trading configurations.

2. Begin a conversation with the chatter bot by typing on the standard terminal.

3. The bot will then initiate a question. It will be displayed on to the terminal window. This will almost always be small talk at the beginning of the conversation. Answer the questions the bot asks by typing back into the terminal window.

4. The conversation will be lead by the bot, i.e.,

    - the bot will either ask the question to which the customer will respond (when did you put in the buy order?), or

    - the bot will instruct the customer to perform some action (change the configuration of the account) to which the customer will answer affirmatively that he / she has completed the action, or answer negatively that he / she is unable to perform the action with a qualifier (I am unable to access the reset password form. I do not have my customer relationship number.) or

    - The bot will ask a question that will require a Yes or No answer.

5. The responses of the customer has to be an exact match with the expected answer in the goal fulfillment map, irrespective of the response that the customer choses. For example, in response to a query from the bot: "Do you remember what kind of orders did you place?"

    - The customer can either answer negatively "No, I do not remember" or
    - The customer can answer "Yes, they were buy orders" or "Buy orders" or "Yes, buy orders"
    - The customer can answer "Yes, they were sell orders" or "Sells orders" or "Yes, sell orders"

    But the customer cannot answer "Very unlikely they were buy orders, but I am not really sure". This is because sentence similarity hasn't been implemented in this architecture. Sentence similarity is the area of research that reduces a range of semantically similar sentences into a root sentence [59, 57]. Hence for this dissertation, the responses need to have the exact words with only a slight change in grammar.

6. The transcript of the conversation is written to a file, and is tagged with the customer utterance and bot utterance. These transcripts can then be analyzed.

Using these steps, 48 conversations were generated. The breakdown was as follows: 8 Informational Conversations, 8 Procedural Conversations, 16 Troubleshooting Conversations, and 16 Dispute Resolution Conversations.

Chapter 7

Results and Discussion

In the introduction it was mentioned that the Loebner Prize is a competition in which chatter bot designers compete to create the chatter bot that can be best passed off as human [48]. It is based on the standard Turing test [93], where a chatter bot and a human operator using a standard computer terminal hold textual conversations with a human judge. Based on the utterances, the judge decides which one of the two is the human and which one is the computer. As elaborated in chapter 1, the winning chatter bots in the Loebner Prize competition do very well for pairs of utterance exchanges, but perform poorly for longer conversations where the context is maintained across several utterance-exchange pairs. This entire dissertation described an architecture to overcome this problem. It must now be asked what is the best method is to measure how successful has my approach been? This chapter discusses evaluation methodology, results, and insights from the results.

7.1 Evaluation Methodology

It has been shown throughout this dissertation, that conversations are inherently noisy but yet principled processes. While there is a lot of randomness that can be approximately stochastically as shown in chapter 5 and 6, conversations also adhere to some conventional principles that are defined by social contracts and specific contexts. It follows that evaluating conversions will essentially be a subjective as well an objective endeavor. This chapter developed a principled way of doing this.

7.1.1 Objective and Subjective Metrics

Chapter 4 developed the idea of using Grice's maxims from the theory of pragmatics to evaluate computer conversations. It was shown in the literature that Grice's maxims, forming the central tenets of the theory of meaning modeling in conversations, was long considered the gold standard for evaluating human conversations. These maxims applicable to customer service conversations on an online electronic trading website, between a human customer and a chatter bot agent are as follows.

* **Quality Maxim:** Agent's responses are factually true, as can be verified by specific evidence from the situational context and domain.

* **Quantity Maxim:** Agents responses are neither too brief, i.e., agent provides too little or incomplete information, nor too verbose, i.e., agent provides too much or superfluous information with respect to the situational context and domain.

* **Relation Maxim:** Agent's responses are relevant to the topic of the conversation with respect to the situational context and domain.

* **Manner Maxim:** Agents responses avoid ambiguity and obscurity, i.e., direct and straightforward with respect to the situational context and domain.

For example, consider a situation where an insurance company offers Home Insurance for USD 1,000 per month, Car Insurance for USD 100 per month, and Combined Home and Car Insurance for USD1,020 per month. Here is how the following bits of conversation between a human customer and a chatter bot agent would be evaluated according to the above Gricean Maxims.

Quality

Human : `How much does the Car Insurance cost?`

Agent: `The Car Insurance costs USD 80 per month.`

Quality Maxim is not satisfied since agent's response is not factually true.

Human : `How much does the Car Insurance cost?`

Agent: `The Car Insurance costs USD 100 per month.`

Quality Maxim is satisfied since agent's response is factually true.

Quantity.

Human : `How much does the Combined Car and Home Insurance cost?`

Agent: `We offer 3 types of insurance plans. If you buy only Home Insurance, it costs USD 1,000 per month and if you buy only Car Insurance, it costs USD 100 per month. But if you buy Combined Insurance you can save money since it costs USD 1,020 per month.`

Quantity Maxim is not satisfied since agent's response is very superfluous and verbose.

Human : `How much does the Car Insurance cost?`

Agent : `The Car Insurance costs USD 100 per month.`

Quantity Maxim is satisfied since agent's response is just right, neither too verbose, nor too laconic.

Relation

Human : `How much does Home Insurance cost?`

Agent: `Our insurance plans are a great deal. Repairing your house after storm damage would cost you more than USD 30,000. But with our insurance all repairs will be covered. Our plans are ranked highest in the state for customer satisfaction.`

Relation Maxim is not satisfied since agent's response is irrelevant to the conversation.

Human : `How much does the Home Insurance cost?`

Agent : `The Home Insurance costs USD 1,000 per month.`

Relation Maxim is satisfied since agent's response is relevant and to the point.

Manner

Human : `Is it cheaper to buy Car and Home Insurance together or separately?`

Agent : `You could save money with either option. Both our Car Insurance and Home Insurance are cheap. The combined Insurance plan is also cheap.`

Manner is unsatisfactory since agent's response is ambiguous and not clear.

Human : `Is it cheaper to buy Car and Home Insurance together or separately?`

`Agent: It is cheaper to buy the Car and Home Insurance together.`

Manner is satisfactory since agent's response is clear and straightforward.

An examination of the maxims will make it clear that while the Quantity Maxim is an objective criterion, since it can be verified directly from the background knowledge as encoded in the Knowledge Engine, the Quality Maxim, Relation Maxim, and Manner Maxim are inherently subjective criteria, and cannot be directly evaluated from either the encoded domain knowledge in the Knowledge Engine, or from everything that has been designed so far. This is an important consideration for designing the evaluation methodology.

Several domain specific criteria were also discussed in chapter 4 for specifically evaluating customer service conversations. These are described below.

* **Percentage of Followup Questions:** Each context in a customer service situation has several followup questions that need to be asked to develop a complete perception of the context. In a good conversation, all the available followup questions need to be asked.

* **Percentage of Successful Resolutions:** The purpose of the chatter bot in this particular domain is to ultimately solve customer problems. Hence, the success ratio achieved by the chatter bot in solving customer issues is an important metric.

* **Number of Coherent Conversation Turns:** In chapter 1, it was shown that the drawback of the prize-winning chatter box from the Loebner competition was that the conversation, while coherent when evaluated in individual utterance-exchange pairs, was not coherent when evaluated across several of

them. Hence, an important metric is to determine the number of utterance-exchange pair over which the chatter bot is coherent.

As an example of application of these domain-specific evaluation criteria, consider the same context and the following conversations between a human customer and a chatter bot agent.

Percentage of Followup Questions

Human : `I would like to open a new trading account? Which plan is the most suitable for me`

Agent: `Would you like to link to an existing margin account or open a separate account?`

Human : `I would like to link to my existing margin account`

Agent: `What is the minimum margin you want for your account?`

Human : `I want a minimum margin of atleast USD 5,000`

Agent: `Then Saver Plan A is the best for your needs.`

* Whether the trading account will link to an existing account or be independent. This information is required to set up the margin allocation.

* The minimum amount of margin the customer wants to trade with. This information is required to set up the transaction channels.

* The frequency with which the customer wants to trade. This information is required to set up the commission structure.

We see that the chatter bot has asked 2 out of 3 followup question to the situational context for a score of 66%.

Percentage of Successful Resolutions

This score will be calculated over series of several conversations in the domain of an electronic trading account website.

$$\frac{Number\ of conversations\ in\ which\ the\ agent\ successfully\ resolves\ the\ issue}{Total\ number\ of\ conversations}$$

Number of Coherent Conversation Turns

Human : `My account is inactive. I am unable to perform any transactions`

Agent: `What is your account number?`

—— Utterance-Exchange Pair 1 ——

Human : `It is CT193830.`

Agent: `What is your mothers maiden name?`

—— Utterance-Exchange Pair 2 ——

Human : `Jones.`

Agent: `Which city were you born in?`

—— Utterance-Exchange Pair 3 ——

Human : `Los Angeles.`

Agent: `Would you like to reactivate the account or close it?`

—— Utterance-Exchange Pair 4 ——

Human : `Why did my account become inactive?`

Agent: `Would you like to reactivate the account or close it?`

—— Utterance-Exchange Pair 5 ——

Human : `I would like to reactivate my account.`

Agent: `Your account has now been reactivated`

—— Utterance-Exchange Pair 6 ——

Human : `Why did my account become inactive?`

Agent: `Your account is active.`

—— Utterance-Exchange Pair 7 ——

An examination of the conversation shows that the conversation goes "off the rails" from Utterance-Exchange Pair 4. Since one of the primary goals of this research is to increase the number of utterance-exchange pair over which the chatter bot can hold a conversation, this is an important evaluation criterion.

Thus, seven distinct metrics have been identified for evaluating the performance of the chatter bot. Out these metrics, four of them, Grice's Quality Maxim, Percentage of Followup Questions, Percentage of Successful Resolutions, and Number of Coherent Conversation Turns can be judges in an objective fashion, since they can be measured or verified simply by examining the conversation transcript, or looked up from the domain knowledge.

The other 3 metrics, Grice's Quantity, Relation, and Manner Maxims cannot be evaluated objectively. They require subjective evaluations. Hence, there needs to be a principled experimental methodology that can combine these evaluation criteria in an scientifically precise and rigorous manner.

7.1.2 Experimental Setup

For the 3 subjective evaluation metrics, i.e, Grice's Quantity, Relation, and Manner Maxims, a panel of human judges was used. The panel consisted of the students in an introductory Computer Science class for non-majors. The class was meant to be a first class in Computer Science, and targeted at freshman students who were not majoring in Computer Science. The class taught basic programming concepts. The exercise for the students was to evaluate artificial conversations based on the three criteria, i.e., Grice's Quantity, Relation, and Manner Maxims.

Since the panel consisted of judges of a relatively similar background, i.e., freshman non Computer Science majors, it is an appropriate control group. Also, since the judges were unlikely to have prior experience with artificial conversation research, or indeed any computer science research, it was unlikely that they could second guess the evaluation exercise and bias the results. Also, since the entire panel of judges was deemed to have sufficient English proficiency, and also be similar to potential users of a system that performed customer service conversations by means of a chatter bot, they were considered appropriate judges to evaluate the results of this research.

The objective of the evaluation exercise was to grade both human conversations, and chatter bot generated conversations. The human conversations were taken directly from the corpus described in chapter 6. The chatter bot conversations were generated as described in chapter 5. A total of 16 human conversations were selected from the corpus and a total of 48 chatter bot conversations were generated for a combined total of 64 conversations. The conversations were comprised of an appropriate mixture from the 4 conversation types described in chapter 5.

For the 16 human conversations, there were 4 procedural conversations, 4 informational conversations, 4 trouble shooting conversations, and 4 dispute resolution conversations. For the 48 chatter bot generated conversations, there were 8 procedural conversations, 8 informational conversations, 16 trouble shooting conversations, and 16 dispute resolution conversations. These numbers are shown in Table 1. Since the Troubleshooting and the Dispute Resolution type of conversations were more interesting and the process of generating them was more involved as described in chapter 5, more of them were included in the analysis as compared to conversations of type Procedural and Informational.

Table 7.1: Distribution of Human Conversations (Natural Conversations) and Chatter Bot generated Conversations (Artificial Conversations). As conversations of type Troubleshooting and Dispute Resolution were more interesting as compared to conversations of type Informational and Procedural, more of the former were included.

Conversation Type	Human	Chatter Bot	Total
Procedural	4	8	12
Informational	4	8	12
Troubleshooting	4	16	20
Dispute Resolution	4	16	20
Total	16	48	64

The set of conversations were divided into 8 different subsets. Each subset consisted of 2 natural conversations and 8 artificial conversations. Each of the 8 subsets were graded by 6 judges. The entire panel had 48 judges and hence each conversation was graded exactly 6 different times. Each judge graded 1 subset consisting of conversation transcripts, out of which 2 were natural conversations and 6 were artificial conversations. Since the class had only 48 students available for the exercise, these were the parameters that resulted.

The grading instructions given to the judges were as follows.

* They were told that they will be shown transcripts of 8 conversations, a combination of natural human conversations and chatter bot generated artificial

conversations. They weren't told which conversations were generated by humans or chatter bots.

* Grice's Quantity, Manner, and Relation Maxims were briefly explained. Examples of conversations that satisfied and did not satisfy these maxims were provided.

* They were asked to grade each conversation against the 3 maxims on a continuous Likert scale. The grade was supposed to indicate to what extent they agreed that the conversations satisfied a maxim on a continuous scale of 0 to 5, with 0 being the worst and 5 being the best.

```
 0----------1----------2----------3----------4----------5
(bad)                                                 (good)
```

* They were asked to guess if the conversation was a human conversation or a chatter bot generated conversation.

The grades from the entire panel for all the subsets of conversation transcripts were collected. Simple statistical analysis was performed on them, and some interesting observations were noticed and are presented next.

7.2 Analysis

To guess whether each conversation was a human conversation or a computer generated conversation, each judge was asked to indicate their choice. They were asked to select exactly one option. Since there were 64 total conversations including 16 natural ones and 48 artificial ones, and each was identified by exactly 6 judges, there was a total of 384 guesses. Out of the 96 possible guesses for the natural conversations, 68 were correctly identified as Human, and 28 were incorrectly identified was a Chatter Bot. For the 288 possible guesses for the artificial conversations, 167 were correctly identified as a Chatter Bot, and 121 wee incorrectly identified as a Human. The numbers are shown in Table 2.

Table 7.2: Identification of who generated the conversation as guessed by the judges. Each conversation was guessed on by exactly 6 judges.

Type	Identified as Human	Identified as Chatter Bot	Total
Natural Conversation	68 (71%)	28 (29%)	96
Artificial Conversation	121 (42%)	167 (58%)	288
Total	189	195	384

Thus for natural conversations, there was a true identification of 70.8% and a false identification of 29.2%. For the artificial conversations, there was a true identification of 58% and a false identification of 42%. Thus the panel of judges was more certain that the natural conversations were generated by a human than they were certain that the artificial conversations were generated by a chatter bot.

As mentioned in the section on experiment design, there were 8 distinct subsets of conversations. Each subset had 2 natural conversations and 6 artificial conversation. In each subset, the first and fourth conversations were human conversations while the rest chatter bot generated. Each conversation had an unique identifier. For example, the conversations in subset 1 had identifiers of 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, and 1.8. Out of these, 1.1 and 1.4 were natural conversations, and 1.2, 1.3, 1.5, 1.6, 1.7, and 1.8 were artificial conversations. Similarly, in subset 2, natural conversations were 2.1 and 2.4, and artificial conversations were 2.2, 2.3, 2.5, 2.6, 2.7, and 2.8.

Table 3 shows the average scores for the natural conversations against Grice's Quantity, Manner, and Relation Maxim. These scores are averaged across 6 judges for each conversation and for each maxim. The raw score given by each judge was on a 0 to 5 continuous Likert scale. These scores were used to normalize each judges

Table 7.3: Average Likert scores for human generated natural conversations. For each conversation, scores are assigned on a continuous 0 to 5 scale and averaged across 6 judges

Conversation ID	Quantity Avg	Manner Avg	Relation Avg
1.1	4	4.2	4.5
1.4	2.9	4.1	4.9
2.1	3.1	3.4	4.5
2.4	4.1	4.5	4.5
3.1	4.1	4.6	4.7
3.4	3.9	4.4	4.4
4.1	4.2	4.7	4.9
4.4	3.9	4.6	4.9
5.1	4.4	4.6	4.6
5.4	3.9	4.7	4.6
6.1	3.9	4.6	4.8
6.4	3.6	4.2	4.6
7.1	4.2	4.6	4.4
7.4	4.4	4.6	4.4
8.1	3.4	4.2	4.4
8.4	3.9	4.6	4.5
Average	3.8	4.4	4.6
Standard Deviation	0.4	0.3	0.2

score for the artificial conversations in the same subset.

For example, in set 1, if a judge assigned scores of 4.3 and 4.1 for the Manner Maxim for conversations 6.1 and 6.4 (both natural) respectively, then the average natural score for this judge would be 4.2 for the Manner Maxim. Now, if this judge assigned scores of 3.8, 3.9, 3.7, 4.1, 4.0, and 3.6 for the Manner Maxim for conversations 6.2, 6.3, 6.5, 6.6, 6.7, and 6.8 (all artificial) respectively, then these scores would be normalized by the average score for the natural conversations. Thus the scores for this judge for artificial conversations 6.2, 6.3, 6.5, 6.6, 6.7, and 6.8 would be 0.90, 0.93, 0.88, 0.98, 0.95, and 0.86 respectively. Similarly, the normalized scores for the artificial conversation given by all 6 judges would be calculated. The average of these 6 normalized scores would be the final score of the artificial conversation for the Manner Maxim. Similarly, the average normalized scores would be calculated for the Relation Maxim and Quantity Maxim. Table 4 shows the scores for all the artificial conversations.

It is interesting to observe that several artificial conversations have an average normalized score higher than 1.0 for the Quantity Maxim. This would mean that according to the panel of judges the chatter bot does better than a human on the Quantity Maxim. This observation should however come with a caveat. Since the human conversations were taken from actual customer service interactions between a human customer and a human customer service representative, most conversations tend to be on the verbose side. The Knowledge Engine for the chatter bot was programmed to consist of a more concise sentence vocabulary. Hence, the judges could have felt that the human conversations were too verbose and marked down the score for the Quantity Maxim. The averaged normalized scores for artificial conversations for the Manner Maxim and Relation Maxim tend me to be lower than 1.00. The average score across all artificial conversations for the Quantity Maxim, Manner Maxim, and Relation Maxim is 1.05, 0.85, and 0.85 respectively. For statistical significance, the one-sample Student's t-test is calculated. It turns out that the difference between the natural and the artificial conversations is statistically significant for the Quantity, Manner, and Relation Maxims.

For the objective evaluation criterion, Table 5 shows the scores for all the conversations. For the Quality Maxim, the Likert scores for the natural conversations were 5, since it is assumed that the human customer service representative is always telling the truth. For the artificial conversations, the Likert score is the proportion of truths told by the chatter bot to the number of times it needed to make a statement that could be true or false according to the encoded domain knowledge in the corpus. The normalized scores are shown in Table 5. The average performance across all artificial conversations is 0.80. According to the one-sampled Student's t-test, the difference between the natural and artificial conversations is statistically significant for the Quality Maxim.

The number of coherent turns of utterance-exchange pairs of the conversation until the conversations goes "off the rails" is also presented in Table 5. This indicated the number of coherent turns before the conversation veers off for the first time. Several such conversations may veer back on course, but those are not counted. The objective is to keep the conversation coherent as long as possible. The average number of coherent turns of utterance-exchange pairs across all artificial conversations is 5.88. This is an improvement over the winning chatter bots from the Loebner Prize competition, which were coherent only across 1 or 2 utterance exchange pairs.

For the follow up percentage, the fraction of follow up sub-contexts (either ordinary utterances or questions) for every context is calculated and presented in Table 5. The chatter bot follows up correctly around 86% of the time. For the human generated natural conversations, this figure is assumed to be 100%. According to the one-sampled Student's t-test, the difference between the natural and artificial conversations is statistically significant for the follow up percentage.

The last column in Table 5 indicates whether the issue was sussesfully resolved by the chatter bot. Across all artificial conversations, the issue was successfully resolved 89% of the times (42 out of 48 artificial conversations).

It is interesting to observe how the scores for each of the evaluation criteria correlate with the success of the artificial conversation in resolving issues. Figure 7.1 shows the relationship between the Quality Maxim and the success of the 48 artificial conversations. The figure indicates that success is highly correlated with the Quality Maxim.

Figure 7.2, shows that the correlation between the success of the artificial conversation and the Quantity Maxim is fairly low. There are some unsuccessful con-

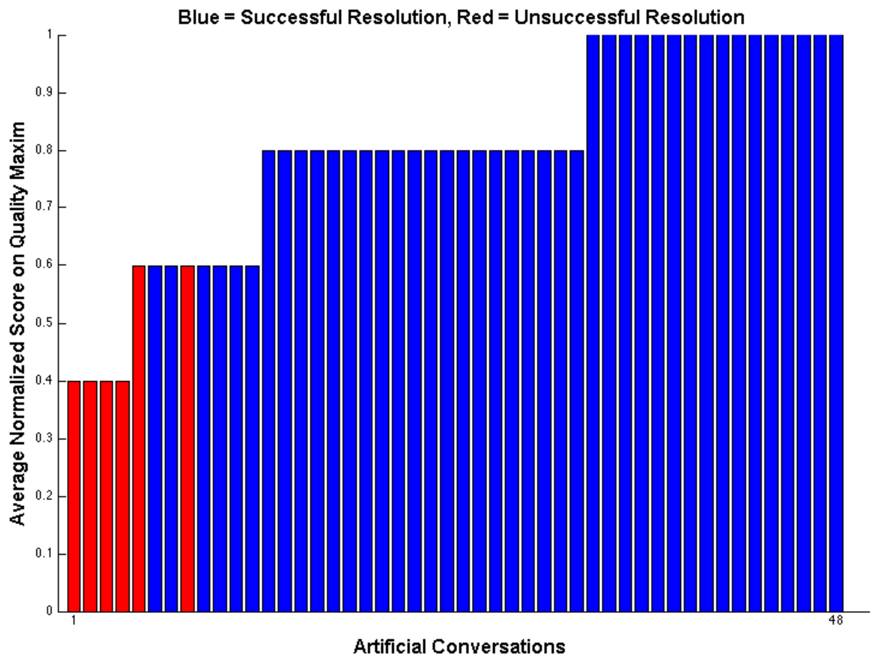

Figure 7.1: Relationship between successful and unsuccessful resolutions in the artificial conversations and the average normalized score for the Quality Maxim. The x-axis shows each of the 48 artificial conversations, and the y-axis shows the average normalized score. Blue indicates a successful resolution and red indicates an unsuccessful resolution.

versations that have a score higher than 1, i.e, the judges felt that the chatter bot did better than a human, for the Quantity Maxim.

Figure 7.3 shows that success is fairly correlated with the Relationship Maxim. Figure 7.4 shows that success is highly correlated with the Manner maxim.

Interestingly, Figure 7.5 indicates that success if perfectly correlated with the follow up percentage, i.e., the number of follow up sub-contexts in the artificial conversation that the chatter bot can correctly address, calculated across all contexts int he conversation.

Figure 7.6 also shows that success is highly correlated with the number of coherent

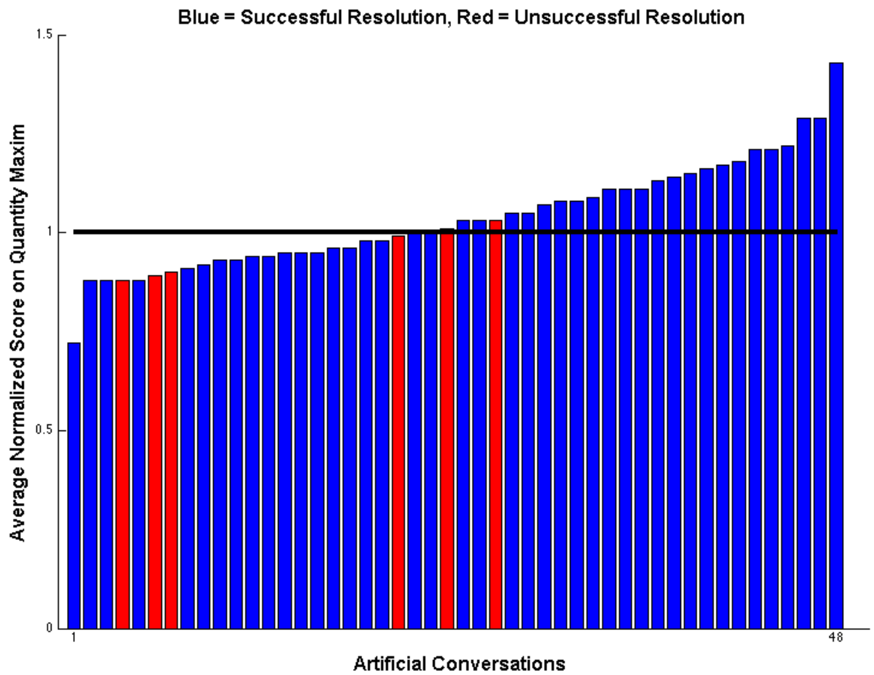

Figure 7.2: Relationship between successful and unsuccessful resolutions in the artificial conversations and the averaged normalized score for the Quantity Maxim. The x-axis shows each of the 48 artificial conversations, and the y-axis shows the average normalized score. The black horizontal line indicates the score for the human generated natural conversations, that is by definition 1.00. Blue indicates a successful resolution and red indicates an unsuccessful resolution.

turns. An artificial conversation with higher number of coherent turns is more likely to successfully resolve the issue.

A key take away from these figures, is that the Quantity Maxim could be perceived to be less important as compared to the Quality, Relation, and Manner Maxims by a panel of judges. It is also possible that this is an artifact of how the goal-fulfillment maps are encoded in this dissertation. Since, the natural conversations are actual human conversations, they are more engaging and verbose. The artificial conversations are programmed to be very precise and to the point. Hence, it is possible that this has been perceived as better fulfillment of the Quantity Maxim

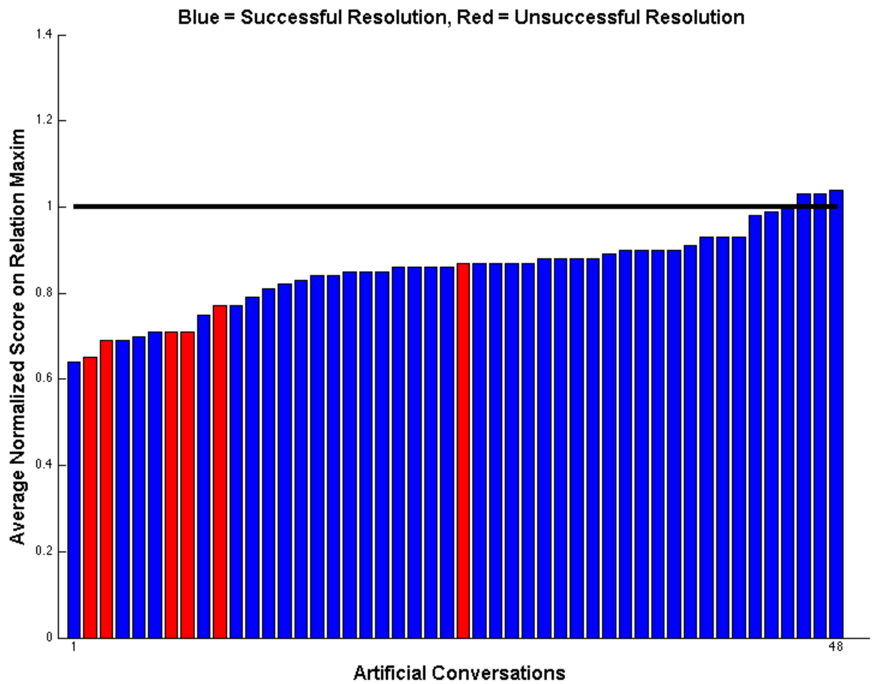

Figure 7.3: Relationship between successful and unsuccessful resolutions in the artificial conversations and the averaged normalized score for the Relation Maxim. The x-axis shows each of the 48 artificial conversations, and the y-axis shows the average normalized score. The black horizontal line indicates the score for the human generated natural conversations, that is by definition 1.00. Blue indicates a successful resolution and red indicates an unsuccessful resolution.

by the panel of judges. Increasing the fraction of follow up questions addressed and the number of coherent turns in the artificial conversation is important for successful resolutions. This should be a key consideration of chatter bot design.

Figure 7.4: Relationship between successful and unsuccessful resolutions in the artificial conversations and the averaged normalized score for the Manner Maxim. The x-axis shows each of the 48 artificial conversations, and the y-axis shows the average normalized score. The black horizontal line indicates the score for the human generated natural conversations, that is by definition 1.00. Blue indicates a successful resolution and red indicates an unsuccessful resolution.

Table 7.4: Average normalized score for chatter bot generated artificial conversations. For each conversation, the score is normalized by the average score given to the natural conversation by the same judge. The normalized score is then averaged across the 6 judges who evaluated the conversation.

Conversation ID	Quantity Normalized	Manner Normalized	Relation Normalized
1.2	1.29	0.99	0.88
1.3	1.21	0.83	0.99
1.5	1.43	0.95	1.03
1.6	1.09	0.69	0.90
1.7	1.17	0.72	1.03
1.8	1.11	0.99	0.98
2.2	0.91	1.06	0.93
2.3	1.16	0.98	0.93
2.5	1.13	1.01	0.84
2.6	0.88	1.06	1.00
2.7	1.08	0.90	0.93
2.8	1.29	1.16	1.04
3.2	1.21	0.87	0.89
3.3	1.05	0.87	0.82
3.5	0.88	0.87	0.91
3.6	0.89	0.72	0.87
3.7	0.88	0.61	0.69
3.8	0.90	0.63	0.65
4.2	1.11	0.87	0.81
4.3	1.07	0.83	0.71
4.5	0.93	0.77	0.75
4.6	0.72	0.81	0.64
4.7	0.88	0.85	0.79
4.8	1.01	0.79	0.77
5.2	0.95	0.85	0.86
5.3	1.03	0.89	0.88
5.5	1.03	0.83	0.70
5.6	0.93	0.85	0.86
5.7	0.94	0.81	0.86
5.8	0.98	0.89	0.86
6.2	1.14	0.97	0.88
6.3	1.18	0.90	0.88
6.5	1.05	0.88	0.84
6.6	0.95	0.83	0.69
6.7	1.03	0.62	0.71
6.8	1.00	0.90	0.77
7.2	0.96	0.81	0.90
7.3	1.00	0.83	0.90
7.5	0.99	0.53	0.71
7.6	0.95	0.86	0.87
7.7	0.96	0.81	0.90
7.8	0.92	0.90	0.87
8.2	0.98	0.94	0.85
8.3	1.08	0.83	0.87
8.5	1.22	0.76	0.83
8.6	1.11	0.89	0.85
8.7	0.94	0.94	0.85
8.8	1.15	0.71	0.87
Average	1.05	0.85	0.85
Standard Deviation	0.13	0.12	0.10
Student's t-test	$p < 0.05$	$p < 0.0001$	$p < 0.0001$
Statistically Significant?	Yes	Extremely	Extremely

Table 7.5: Scores for the objective metrics, i.e, Grice's Quality Maxim, Number of Coherent Turns, and Successful resolutions. The quality Maxim is normalized by the score for the natural conversations, which is always 5.

Conversation ID	Quality Normalized	Coherent Turns	Followup Percentage	Successful?
1.2	1	5	0.67	YES
1.3	1	6	1	YES
1.5	1	6	1	YES
1.6	1	6	1	YES
1.7	0.8	5	1	YES
1.8	1	8	1	YES
2.2	1	5	0.67	YES
2.2	0.8	5	0.67	YES
2.5	1	7	0.86	YES
2.6	1	6	1	YES
2.7	1	7	1	YES
2.8	1	7	1	YES
3.2	1	8	1	YES
3.3	0.8	5	0.673	YES
3.5	1	7	1	YES
3.6	0.6	4	0.5	NO
3.7	0.4	3	0.29	NO
3.8	0.4	3	0.38	NO
4.2	0.8	7	1	YES
4.3	0.8	6	1	YES
4.5	0.6	5	0.8	YES
4.5	0.6	5	0.8	YES
4.7	0.8	6	1	YES
4.8	0.6	5	0.5	NO
5.2	0.8	6	0.75	YES
5.3	1	8	1	YES
5.5	0.8	6	1	YES
5.6	0.8	6	1	YES
5.7	0.8	6	1	YES
5.8	0.8	6	1	YES
6.2	1	7	1	YES
6.3	0.8	6	1	YES
6.5	0.8	6	1	YES
6.6	0.6	5	0.75	YES
6.7	0.4	4	0.4	NO
6.8	0.8	6	1	YES
7.2	0.8	6	1	YES
7.3	0.8	6	0.75	YES
7.5	0.4	4	0.4	NO
7.6	0.8	6	1	YES
7.7	0.8	6	1	YES
7.8	0.8	6	0.67	YES
8.2	1	8	1	YES
8.3	0.6	5	0.67	YES
8.5	0.6	5	1	YES
8.6	0.8	7	1	YES
8.6	1	8	1	YES
8.8	0.6	6	1	YES
Average	0.80	5.88	0.86	NA
Standard Deviation	0.18	1.12	0.21	NA
Student's t-test	$p < 0.0001$	NA	$p < 0.0001$	NA
Statistically Significant?	Extremely	NA	Extremely	NA

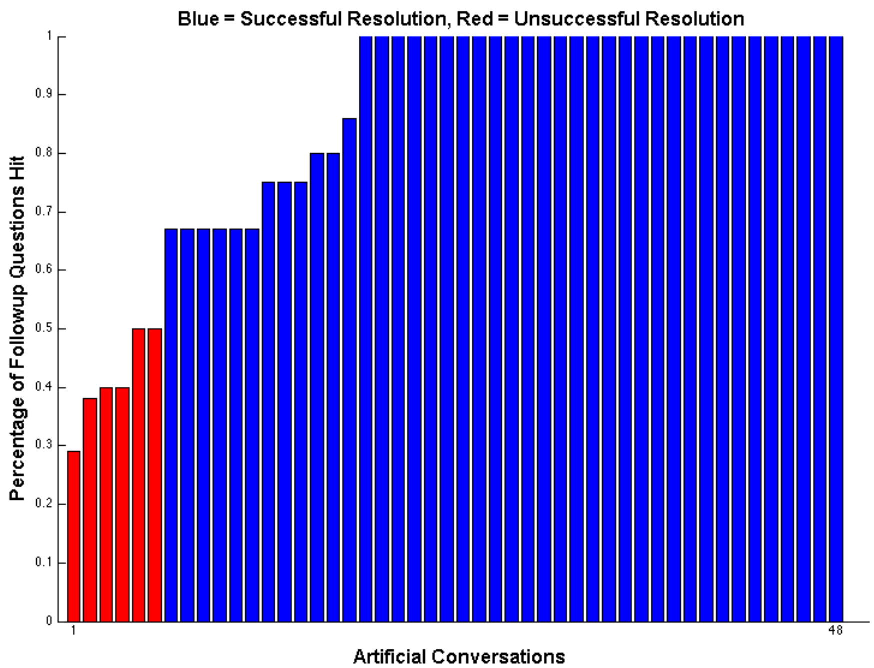

Figure 7.5: Relationship between successful and unsuccessful resolutions in the artificial conversations and the follow up percentage. The x-axis shows each of the 48 artificial conversations, and the y-axis shows the average normalized score. Blue indicates a successful resolution and red indicates an unsuccessful resolution.

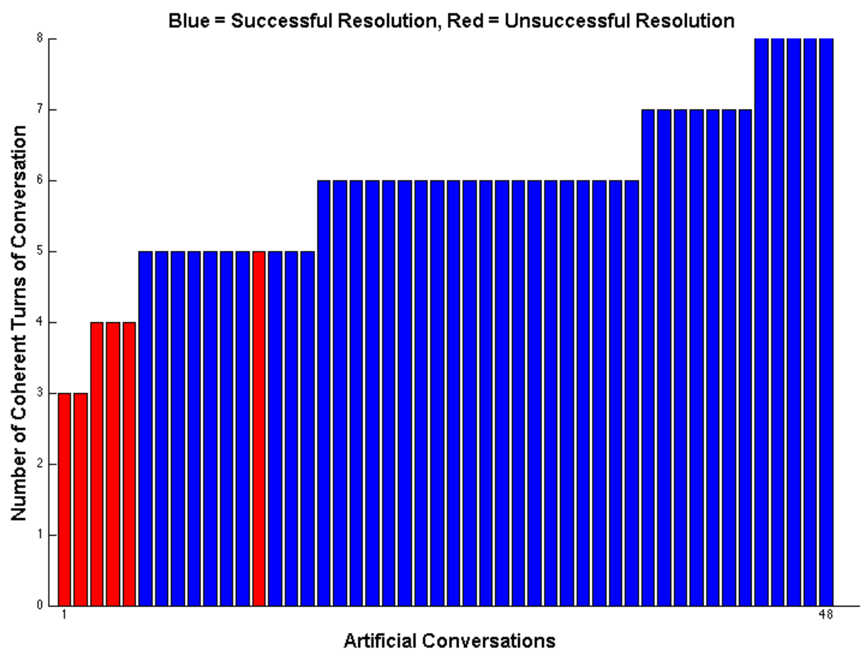

Figure 7.6: Relationship between successful and unsuccessful resolutions in the artificial conversations and the number of coherent turns.. The x-axis shows each of the 48 artificial conversations, and the y-axis shows the average normalized score. Blue indicates a successful resolution and red indicates an unsuccessful resolution.

Chapter 8

Conclusions and Future Work

This research set out to create a chatter bot architecture that would overcome the limitations of state of the art chatter bots in generating higher quality artificial conversations. It has been reasonably successful at doing that as demonstrated in chapter 7.

The criteria selected to evaluate artificial conversations was determined according to the current literature. Conversation theory and the theory of pragmatics have been well established scientific field for several decades. It follows that since the our goal was to enable chatter bots to generate more human-like conversations, using the same criteria that has been used to evaluate human conversations by psycholinguists, pragmaticists, and conversation theorists is appropriate for evaluating artificial conversations as well.

The inductive bias necessary for the research limited conversations to the domain of customer service conversations between a human customer and a chatter bot. A corpus of customer service chat transcripts between a human customer and a human customer service representative was available. This was used to train language models that were used in the architecture to generate artificial conversations.

In keeping with the domain specific nature of this research, evaluation metrics were defined suitable for customer service situations. Natural conversations from the corpus were used as a benchmark to compare the performance of the chatter bot in generating artificial conversations. The results obtained were satisfactory, though there are several caveats and ample scope for improvements.

8.1 Unique contributions

The conclusions next present several unique contributions from my research.

1. A modular, robust, and scalable architecture was presented for generating artificial conversations. The design was based on a plug-and-play philosophy in which the functions of the Knowledge Engine and the Conversation Engine could be implemented in multiple ways using multiple techniques. One such specific technique, implementing the knowledge engine using goal fulfillment maps, was demonstrated. Another specific technique, implementing the conversation engine using stochastic finite state automata, was also demonstrated.

2. A specific set of evaluation criteria was defined for evaluating artificial conversations. A technique to use natural conversations to benchmark the quality of artificial conversations was also demonstrated. The evaluation criteria included both objective and subjective metrics, and were applicable to both general-purpose conversations and purpose-driven domain specific and context specific situational conversations. These criteria were grounded in the scientific literature used to evaluate natural conversations by humans.

3. This research is unique in that it demonstrates an effective method to combine content semantics and pragmatic semantics. A good conversations depends on both semantically relevant underlying process, as well as being grounded in a

set of facts from a knowledge base. Several approaches in literature focus on either building robust principled knowledge representation techniques for conversations or developing new semantic modeling techniques for conversations. This is the first approach that combines content semantics in the form of a knowledge engine and pragmatic semantics in the form a conversation engine to generate high quality artificial conversations.

8.2 Limitations of the research

There are several limitations of this research:

1. Since the evaluation methodology involved several subjective metrics, judges were needed to grade the quality of the artificial conversations against these metrics. This introduced statistical noise and biases in the evaluation. Although steps were taken to eliminate these biases, this was limited by the small number of judges.

2. Also, the process of grading by human judges introduced a feedback lag in the iterative research process. It was not feasible to tune specific parameters of the conversation model and observe how that influenced the results, since that process would need to go through a long evaluation process by further judges. There was no mechanism for immediate feedback.

3. There was no analysis on how well the methods described in this research could generalize to non-specific conversations not attuned to a narrow situational context. Indeed this is still an open question.

8.3 Possible future directions

There are several exciting directions in which this work can be extended. Some potential directions are described below.

1. An automated evaluation mechanism that relies on a set of objective metrics that can be quantitatively benchmarked might overcome the limitation of slow non-immediate feedback.

2. Incorporating richer knowledge representation and retrieval techniques, such as ontologies might make the architecture work even with less situation specific contextual conversations.

3. This research considered only four types of conversations, i.e., Procedural, Informational, Troubleshooting, and Dispute Resolution. Other types of conversations can be defined and the modeling and analysis can be extended to these types.

4. The conversations were modeled using stochastic finite state automata, which worked well in narrow situational contexts. More formally richer modeling frameworks like Partially Observable Markov Decision Processes (POMDPs) [40] might be useful for modeling wider ranges of contexts. While POMDPs suffer from challenges of computational intractability, there are are several possible approximation techniques that can deal with intractability. The Policy Learning family of algorithms from Reinforcement Learning are a potential solution for this type of modeling and approximation.

5. The implementation of a sentiment detector module in the Chat Interface will enable the chatter bot to account for polarity of sentiment in the conversation and help generate more realistic artificial conversations incorporating the emotions and attitudes.

6. The implementation of a sentence similarity module in the Chat Interface will enable the chatter bot handle of wide range of specific utterances that convey the same meaning to the context in the conversation. A sentence similarity module will reduce an utterance to it's root sentence [59, 56, 58].

7. The modeling of conversation failure and recovery mechanisms will enable the chatter bot to better handle the conversations that fail according to the stochastic automata. This can be achieved by computationally modeling the concept of conversation repair.

8. The modeling of conversations across multiple contexts will enable the chatter bot to generate artificial conversations that handle more than one context simultaneously.

References

[1] W P Alston, *Illocutionary acts and sentence meaning.*, Cornell University Press., Ithaca, N.Y., 2000.

[2] John Austin, *How to do things with words.*, Oxford University Press, 1962.

[3] Daniel G. Bobrow, Ronald M. Kaplan, Martin Kay, Donald A. Norman, Henry Thompson, and Terry Winograd, *Gus: A frame-driven dia—og system*, Artificial Intelligence **8** (1977), 155–173.

[4] Rollo Carpenter, *Jabberwacky*, 2007.

[5] Joyce Chai, Shimei Pan, and Michelle Zhou, *Mind: A context-based multimodal interpretation framework in conversational systems*, Advances in Natural Multimodal Dialogue Systems. **30** (2005), no. 5, 265–285.

[6] Chayan Chakrabarti, *Enriching chatter bots with semantic conversation control*, Twenty-Sixth AAAI Conference on Artificial Intelligence Doctoral Consortium (Toronto, Ontario.), AAAI Press, July 2012.

[7] Chayan Chakrabarti and George Luger, *A semantic architecture for artificial conversations*, The 13th International Symposium on Advanced Intelligent Systems (Kobe, Japan), IEEE Press, November 2012.

[8] ———, *A framework for simulating and evaluating artificial chatter bot conversations*, The 26th International Florida Artificial Intelligence Research Society Conference (St. Pete Beach, FL), AAAI Press, May 2013.

[9] Jean-Pierre Changeux, *Conversations on mind, matter, and mathematics*, Princeton University Press, December 1998.

[10] Herbert Clark, *Using language*, Cambridge University Press, Cambridge, 1996.

[11] David Clarke, *Language and action, a structural model of behavior*, Pergamon Press, 1983.

[12] Kenneth Colby, *Idiolectic language-analysis for understanding doctor-patient dialogues.*, IJCAI (1973), 278–284.

[13] Robert Craig and Karen Tracy, *Conversational coherence*, vol. 2, Sage Publications, 1983.

[14] Angelo Dalli, Yunqing Xia, and Yorick Wilks, *Fasil email summarisation system*, Computational Linguistics, 2004.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the em algorithm*, Journal Of The Royal Statistical Society **Series B** (1977).

[16] Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Michael Seltzer, Geoff Zweig, Xiaodong He, Jason Williams, Yifan Gong, and Alex Acero, *Recent advances in deep learning for speech research at microsoft*, IEEE International Conference on Acoustics, Speech, and Signal Processing, 2013.

[17] Shona Douglas, Deepak Agarwal, Tirso Alonso, Robert Bell, Mazin Rahim, Deborah F. Swayne, and Chris Volinsky, *Mining customer care dialogs for "daily news"*, IEEE Transactions on Speech and Audio Processing, Special Issue on Data Mining of Speech, Audio and Dialog, vol. 13, 2005, pp. 652 – 660.

[18] Mohan Embar, *Chip vivant*, 2013.

[19] E. Filisko and S. Seneff, *A context resolution server for the galaxy conversational systems*, Proc. Eurospeech, September 2003.

[20] Mark Fišel, *Machine learning techniques in dialogue act recognition*, Estonian Papers in Applied Linguistics, 2007.

[21] Aldo Gangemi, *Ontology design patterns for semantic web content*, Springer-Verlag Berlin, 2005, pp. 262–276.

[22] Harold Garfinkel, *Studies in ethnomethodology*, Prentice Hall, Englewood Cliffs NJ, 1967.

[23] Robby Garner, *The turing hub as a standard for turing test interfaces*, Parsing the Turing Test (2009), 319–324.

[24] Gartner, *Organizations that integrate communities into customer support can realize cost reductions of up to 50 percent*, Gartner Customer 360 Summit 2012 (2012).

[25] M. Gasic, C. Breslin, M. Henderson, D. Kim, M. Szummer, B. Thomson, P. Tsiakoulis, and S. Young, *Pomdp-based dialogue manager adaptation to extended domains*, SigDial Metz France, 2013.

[26] Jonathan Ginzburg, *Semantics for conversation*, CSLI Publications, King's College, London, 2008.

[27] Erving Goffman, *The interaction order. american sociological review*, 1983.

[28] G GoldKuhl, *Conversational analysis as a theoretical foundation for language action aproaches?*, Proceedings of the 8th international working conference on the languageaction perspective on communication modelling. (Tilburg, The Netherlands.) (H Weigand, G GoldKuhl, and A de Moor, eds.), 2003.

[29] T E Goldsmith and D M Davenport, *Assessing structural similarity of graphs*, Pathfinder associative networks: studies in knowledge organization (1990), 75–87.

[30] Paul Grice, *Meaning*, The Philosophical Review **66** (1957), no. 3.

[31] ______, *Logic and conversation*, Syntax and Semantics **3** (1975), 41–58.

[32] ______, *Studies in the way of words*, Harvard University Press, 1989.

[33] Tom Gruber, *A translation approach to portable ontology specifications*, Knowledge Acquisition **5** (1993), 199–220.

[34] Taher Haveliwala, *Search facilities for internet relay chat*, Joint Conference on Digital Libraries (Portland, Oregon), July 2002.

[35] M. Henderson, B. Thomson, and S. Young, *Deep neural network approach for the dialog state tracking challenge*, SigDial Metz FranceMetz France, 2013.

[36] Eric Horvitz and Tim Paek, *A computational architecture for conversation*, Tech. report, Microsoft Research, 2000.

[37] Minqing Hu and Bing Liu, *Mining and summarizing customer reviews*, Proceedings of KDD, 2004.

[38] Dan Jurafsky, Liz Shriberg, and Debra Biasca, *Switchboard swbd-damsl shallow-discourse-function annotation coders manual*, Tech. report, University of Colorado - Boulder, February 1997.

[39] Daniel Jurafsky and James H. Martin, *Speech and language processing*, 2 ed., Pearson Prentice Hall, May 2008.

[40] L.P. Kaelbling, M.L. Littman, and A.R. Cassandra, *Planning and acting in partially observable stochastic domains*, Artificial Intelligence Journal (1998), no. 101, 99–134.

[41] E. Levin, S. Narayanan, R. Pieraccini, K. Biatov, E. Bocchieri, G. Di Fabbrizio, W. Eckert, S. Lee, A. Pokrovsky, M. Rahim, P. Ruscitti, and M. Walker, *The att-darpa communicator mixed-initiative spoken dialog system*, ICSLP, 2000.

[42] Xiangyang Li and Qiang Ji, *Active affective state detection and user assistance with dynamic bayesian networks*, IEEE Transactions on Systems, Man, and Cybernetics, vol. 35, 2005.

[43] Y Li, *Amethod for measuring sentence similarity and its application to conversational agents*, Florida Artificial Research Society Conference, 2004.

[44] Bing Liu, *Sentiment analysis and subjectivity*, 2 ed., 2010.

[45] Bing Liu, Minqing Hu, and Junsheng Cheng, *Opinion observer: Analyzing and comparing opinions on the web*, Proceedings of WWW, 2005.

[46] John Local, *Phonetic detail and the organisation of talk-in-interaction*, Proceedings of the XVIth International Congress of Phonetic Sciences (2007).

[47] George Luger, *Artificial intelligence: Structures and strategies for complex problem solving*, 6 ed., Addison Wesley, March 2008.

[48] Michael Mauldin, *Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition*, Proceedings of the Eleventh National Conference on Artificial Intelligence (Seattle, Washington), AAAI Press, August 1994.

[49] Manish Mehta and Andrea Corradini, *Developing a conversational agent using ontologies*, HCI'07 Proceedings of the 12th international conference on Human-computer interaction: intelligent multimodal interaction environments, Springer-Verlag, 2007.

[50] Angeliki Metallinou, Dan Bohus, , and Jason D. Williams, *Discriminative state tracking for spoken dialog systems*, Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL), Sofia, Bulgaria, 2013.

[51] Jacob L. Mey, *Pragmatics: An introduction.*, 2 ed., Oxford: Blackwell, 2001.

[52] D. Michie and C. Sammut, *Infochattm scripters manual.*, Convagent Ltd, Manchester, 2001.

[53] Gilad Mishne, David Carmel, Ron Hoory, Alexey Roytman, and Aya Soffer, *Automatic analysis of call-center conversations*, CIKM '05 Proceedings of the 14th ACM international conference on Information and knowledge management, ACM, Octomer - November 2005.

[54] Cristian Moldovan, Vasile Rus, and Arthur Graesser, *Automated speech act classification for online chat*, The 22nd Midwest Artificial Intelligence and Cognitive Science Conference, April 2011.

[55] Marek Obitko, *Translations between ontologies in multi-agent systems*, Ph.D. thesis, Czech Technical University, Prague, Czech, 2007.

[56] K. O'Shea, Z. Bandar, and K. Crockett, *A novel approach for constructing conversational agents using sentence similarity measures*, World Congress on Engineering, International Conference on Data Mining and Knowledge Engineering, 2008, pp. 321–326.

[57] ———, *A semantic- based conversational agent framework*, The 4th International Conference for Internet Technology and Secured Transactions (ICITST-2009), Technical Co- Sponsored by IEEE UK?RI Communications Chapter (London), November 2009, pp. 92–99.

[58] ———, *Towards a new generation of conversational agents using sentence similarity*, Advances in Electrical Engineering and Computational Science, Lecture Notes in Electrical Engineering **39** (2009), 505–514.

[59] K. O'Shea, Z. Bandar, K. Crockett, and D. Mclean, *A comparative study of two short text semantic similarity measures*, Lecture Notes on Artificial Intelligence **4953** (2004), 172.

[60] Karen O'Shea, Zuhair Bandar, and Keeley Crockett, *A conversational agent framework using semantic analysis.*, International Journal of Intelligent Computing Research (IJICR) **1** (2010), no. 1/2.

[61] Ozcan Ozyurt and Cemal Kose, *Chat mining: Automatically determination of chat conversations' topic in turkish text based chat mediums*, Expert Systems with Applications, vol. 37, 2010, pp. 8705–8710.

[62] Tim Paek and Eric Horvitz, *Conversation as action under uncertainty*, Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI), 2000, pp. 455–464.

[63] Alexander Pak and Patrick Paroubek, *Twitter as a corpus for sentiment analysis and opinion mining*, Proceedings of LREC., 2010.

[64] Bo Pang and Lillian Lee, *Opinion mining and sentiment analysis*, Foundations and Trends in Information Retrieval **2** (2008), no. 1-2, 1–135.

[65] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan, *Thumbs up? sentiment classification using machine learning techniques*, Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002, pp. 79–86.

[66] Bo Pang, Lillian, and Lee, *Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales*, Proceedings of the Association for Computational Linguistics (ACL), pp. 115–124.

[67] G Pask, *Conversation theory*, Elsevier., 1976.

[68] J. Polifroni and S. Seneff, *Galaxy-ii as an architecture for spoken dialogue evaluation*, LREC, May 2000.

[69] Simon Polovina and John Heaton, *An introduction to conceptual graphs*, AI Expert (1992), 36–43.

[70] Anita Pomerantz, *Agreeing and disagreeing with assessments: Some features of preferred/dispreferred turn shapes.*, Structures of Social Action: Studies in Conversation Analysis (1984).

[71] M F Porter, *An algorithm for suffix stripping*, Program **14** (1980), no. 3, 130–137.

[72] Lawrence R. Rabiner, *"a tutorial on hidden markov models and selected applications in speech recognition*, Proceedings of the IEEE **77** (1989), no. 2, 257–286.

[73] Verena Rieser and Oliver Lemon, *Reinforcement learning for adaptive dialogue systems: A data-driven methodology for dialogue management and natural language generation*, Springer, 2013.

[74] Shourya Roy and L Venkata Subramaniam, *Automatic generation of domain models for call centers from noisy transcriptions*, Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL (Sydney), 2006, pp. 737–744.

[75] H Sacks, E Schegloff, and G Jefferson, *A simplest systematics for the organization of turn-taking for conversation*, LAnguage **50** (1974), 696–735.

[76] C Sammut, *Managing context in a conversational agent*, Electronic Transactions on Artificial Intelligence **3** (2001), no. 7, 1–7.

[77] Ayse Pinar Saygin and Ilyas Ciceklib, *Pragmatics in human-computer conversation*, Journal of Pragmatics **34** (2002), 227–258.

[78] Emanuel A. Schegloff, *Sequence organization in interaction: A primer in conversation analyis*, vol. 1, Cambridge University Press., 2007.

[79] John Searle, *Indirect speech acts*, Speech Acts, ch. 3, pp. 59–82, Academic Press, New York.

[80] ______, *Speech acts*, Cambridge University Press, 1969.

[81] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, and V. Zue, *Galaxy-ii: A reference architecture for conversational system development*, Proc. ICSLP, Sydney, Australia, November 1998.

[82] Jack Sidnell, *Conversation analysis: An introduction*, Wiley-Blackwell, 2010.

[83] Jack Sidnell and Tanya Stivers, *Handbook of conversation analysis*, Wiley-Blackwell, 2012.

[84] Benjamin Snyder and Regina Barzilay, *Multiple aspect ranking using the good grief algorithm*, Proceedings of the Joint Human Language Technology/North American Chapter of the ACL Conference (HLT-NAACL), 2007, pp. 300–307.

[85] John F. Sowa, *Conceptual structures: Information processing in mind and machine*, Addison-Wesley, 1984.

[86] ______, *Conceptual graphs summary*, Conceptual Structures: Current Research and Practice (1992), 3–52.

[87] Mark Steyvers and Tom Griffiths, *Probabilistic topic models*, Psychology Press, 2008.

[88] Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer, *Dialogue act modeling for automatic tagging and recognition of conversational speech*, Computational Linguistics **26** (2000), no. 3, 339–373.

[89] Andreas Stolcke, Elizabeth Shriberg, Rebecca Bates, Noah Coccaro, Daniel Jurafsky, Rachel Martin, Marie Meteer, Klaus Ries, Paul Taylor, and Carol Van Ess-Dykema, *Dialog act modeling for conversational speech*, AAAI Technical Report, vol. SS-98-01, 1998.

[90] P Subasic and A Huettner, *Affect analysis of text using fuzzy semantic typing.*, IEEE Transactions on Fuzzy Systems, vol. 9, 2001, pp. 483–496.

[91] Hironori Takeuchi, L Venkata Subramaniam, Tetsuya Nasukawa, and Shourya Roy, *Automatic identification of important segments and expressions for mining of business-oriented conversations at contact centers*, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007, pp. 458–467.

[92] Hironori Takeuchi, L Venkata Subramaniam, Tetsuya Nasukawa, Shourya Roy, and Sreeram Balakrishan, *A conversation-mining system for gathering insights to improve agent productivity*, IEEE Joint Conference on E-Commerce Technology (CEC'07) and Enterprise Computing, E-Commerce and E-Services (EEE'07), 2007, pp. 465–468.

[93] Alan Turing, *Computing machinery and intelligence*, Oxford University Press, 1950.

[94] Peter Turney, *Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews*, Proceedings of the Association for Computational Linguistics, pp. 417–424.

[95] Douglas P. Twitchell, Mark Adkins, Jay F. Nunamaker, and Judee K. Burgoon, *Using speech act theory to model conversations for automated classification and retrieval*, Proceedings of the 9th International Working Conference on the Language-Action Perspective on Communication Modelling (2004).

[96] Marilyn Walker, Lynette Hirschman Y, and John Aberdeen Y, *Evaluation for darpa communicator spoken dialogue systems*, Second International Conference on Language Resources and Evaluation, 2000.

[97] Richard S. Wallace, *The anatomy of a.l.i.c.e.*, Parsing the Turing Test **3** (2009), 181–210.

[98] Joseph Weizenbaum, *Eliza—a computer program for the study of natural language communication between man and machine*, Communications of the ACM **9** (1966), no. 1, 36–45.

[99] Simon Whitehead and Lawrence Cavedon, *Generating shifting sentiment for a conversational agent*, NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, Association for Computational Linguistics, June 2010, pp. 89–97.

[100] Bruce Wilcox, *Rosette*, 2012.

[101] T Winograd, *A language/action perspective on the design of cooperative work*, Human Computer Interaction **3** (2003), no. 1, 3–30.

[102] T Winograd and F Flores, *Understanding computers and cognition: A new foundation for design.*, Ablex Publishing Corporation., Norwood, New Jersey, 1986.

[103] Steve Worswick, *Mitsuku*, 2013.

Appendix

Appendix A

Transcripts of Artificial Conversations

Transcripts of Artificial Conversations

Example transcripts of artificial conversations generated by the chatter bot.

Chatter Bot : `Hello. How can I help you today?`

Customer : `I would like to open a new account for day trading. What are my options?`

Chatter Bot : `Do you have an existing trading account or would you like to open a new one?`

Customer : `I already have a trading account with my bank.`

Chatter Bot : What is the maximum amount of daily margin that you will require?

Customer : Not more than $25,000.

Chatter Bot : Are you going to trade in one or more of futures and options, commodities, foreign exchange, and equities?

Customer : I am primarily interested in futures and options on commodities.

Chatter Bot : The Intermediate Plan 2 account is best suited to your requirements. It provides a daily margin of $30,000. It charges a commission of 1% per trade and has immediate execution.

Customer : Great. How do I get an account on that plan?

Chatter Bot : You can download the plan from the Open New Account > Intermediate Plan 2 menu, and send it in along with all the required documents.

Customer : Thank you very much.

Chatter Bot : You are welcome.

−−−−−−−−−−−−−−− End of Conversation −−−−−−−−−−−−−−−

Customer : `Hello.`

Chatter Bot : `Hello. How can I help you today?`

Customer : `My account shows an incorrect daily margin.`

Chatter Bot : `Please give me your account identification number.`

Customer : `It is DFR1948033.`

Chatter Bot : `Is the margin on your account higher or lower?`

Customer : `It is lower.`

Chatter Bot : `Do you have pending buy orders on your account?`

Customer : `No.`

Chatter Bot : `Do you have pending sell orders on your account?`

Customer : `Yes, but I cancelled them before they were executed.`

Chatter Bot : `When did you cancel them?`

Customer : `One day ago`

Chatter Bot : `It takes 48 hours for cancel instructions to process.`

Customer : `I got it.  Thank you.`

Chatter Bot : `You are welcome.`

– – – – – – – – – – – – – – – End of Conversation – – – – – – – – – – – – – – –

Customer : Hello.

Chatter Bot : Hello. How are you doing today?

Customer : I am doing good. What about you?

Chatter Bot : Very well. Thank you. And what is your name?

Customer : My name is Sid.

Chatter Bot : Hello Sid. How can I help you today?

Customer : I would like to set up my margin for recurring transactions.

Chatter Bot : What is your account number?

Customer : My account number is CX219234.

Chatter Bot : Would you like to have recurring transaction or single transaction?

Customer : I would like to have recurring transactions. I am a day trader.

Chatter Bot : `Would you like to have rolling margin allocation or single allocation?`

Customer : `I would like to have rolling allocations.`

Chatter Bot : `You can now configure your account for day trading mode. I have set the parameters.`

Customer : `Thank you. That was quick.`

Chatter Bot : `You are welcome.`

– – – – – – – – – – – – – – – End of Conversation – – – – – – – – – – – – – – –

Customer : `Hello.`

Chatter Bot : `Hello. How can I help you today?`

Customer : `My account is inactive. I am unable to perform any transactions.`

Chatter Bot : `What is your account number?`

Customer : `It is CT193830.`

Chatter Bot : `What is your mothers maiden name?`

Customer : `Jones.`

Chatter Bot : `Which city were you born in?`

Customer : `Los Angeles`

Chatter Bot : `Would you like to reactivate the account or close it?`

Customer : `Why did my account become inactive?`

Chatter Bot : `Would you like to reactivate the account or close it?`

Customer : `I would like to reactivate my account.`

Chatter Bot : `Your account has now been reactivated.`

Customer : `Why did my account become inactive?`

Chatter Bot : `Your account is active.`

Customer : `OK. Thanks I guess.`

Chatter Bot : `You are welcome.`

– – – – – – – – – – – – – – – End of Conversation – – – – – – – – – – – – – – –

Customer : `Hello.`

Chatter Bot : `Hello. How can I help you today?`

Customer : `I have been wrongly charged fees for a transaction.`

Chatter Bot : `What is your account number?`

Customer : `It is FGT3390.`

Chatter Bot : `What is your mothers maiden name?`

Customer : `Parkinson.`

Chatter Bot : `Which city were you born in?`

Customer : `Ithaca.`

Chatter Bot : `What is the transaction ID for which you have a problem?`

Customer : `TRN2341`

Chatter Bot : `When did you make the transaction?`

Customer : I put in the order on Monday January 21. But I cancelled the order on Thursday January 24 within the 3-??day period. So no fees should have been charged.

Chatter Bot : Transactions usually take 48 to 72 hours to process. Has you transaction been carried out?

Customer : It was carried out.

Chatter Bot : Your transaction went through in 48 hours and hence you have charged the transaction fee.

Customer : Can I do anything to reverse the transaction?

Chatter Bot : Since the transaction is a market transaction, it cannot be reversed.

Customer : Well thanks for nothing then.

Chatter Bot : You are welcome.

– – – – – – – – – – – – – – – End of Conversation – – – – – – – – – – – – – – –

Chatter Bot : `Hello. How can I help you today?`

Customer : `I have a problem with my portfolio.`

Chatter Bot : `Can you please explain the problem?`

Customer : `I wanted to change my allocation. Sell $2,000 worth of the IMF Mutual Fund and put that amount for the margin in my derivatives.`

Chatter Bot : `Are you able to change your allocation?`

Customer : `No.`

Chatter Bot : `Which allocation are you trying to reduce?`

Customer : `Mutual Funds.`

Chatter Bot : `Which allocation are you trying to increase?`

Customer : `Derivatives.`

Chatter Bot : `Are you able to reduce your allocation for Mutual Funds?`

Customer : `No. It doesnt allow me to reduce.`

Chatter Bot : Are you able to increase your allocation for Derivatives?

Customer : Since I have been unable to reduce my allocation from Mutual Funds, I have been unable to increase my allocation to Derivatives.

Chatter Bot : You need to decrease your allocation before you can make an increase in margin.

Customer : But I am unable to decrease my allocation.

Chatter Bot : Which portfolio are you unable to decrease allocation for?

Customer : Mutual Funds.

Chatter Bot : Are your Mutual Funds fixed deposits?

Customer : Yes, they are fixed deposits.

Chatter Bot : You cannot reallocate from fixed deposits before the fixed deposit term expires.

Customer : Oh I see. Thanks anyway.

– – – – – – – – – – – – – – End of Conversation – – – – – – – – – – – – – –

Customer : `I need help in placing an order.`

Chatter Bot : `Have you logged into your account?`

Customer : `Yes.`

Chatter Bot : `Are you on your account home page?`

Customer : `Yes, I am on my home page. I see me name on the top left hand corner.`

Chatter Bot : `What kind of order would you like to place?`

Customer : `I would like to purchase $3,000 worth of stock in ACN.`

Chatter Bot : `Click on the Equity link on your home page.`

Customer : `I did that.`

Chatter Bot : `What is the margin displayed on the Equity page.`

Customer : `$3,331.`

Chatter Bot : `You have sufficient margin to place your order.`

Customer : `So how do I place the order?`

Chatter Bot : `Is it a buy or sell order?`

Customer : `It is a buy order.`

Chatter Bot : `Click on the purchase box. You will be taken to a transaction form where you can fill in the details of your desired order.`

Customer : `Great. Let me try it. Thanks.`

Chatter Bot : `You are welcome.`

– – – – – – – – – – – – – – – End of Conversation – – – – – – – – – – – – – – –

Customer : Hello. I would like to open a new account with eTRADE.

Chatter Bot : Thats great. Let me ask you a few questions that will help us determine the best account plan for you.

Customer : Sure.

Chatter Bot : Is this going to be a new account or do you already have an account with us?

Customer : This will be a new account.

Chatter Bot : What do you primarily want to use the account for?

Customer : I would like to invest in Mutual Funds and stocks.

Chatter Bot : How often do you plan to be trading using your account?

Customer : Not very often.

Chatter Bot : Will you be using your account for day trading?

Customer : No, just for occasionally trading in stocks and buying mutual funds.

Chatter Bot : `What is the approximate amount you plan to invest in initially?`

Customer : `Ill start with $20,000 but might add to it every quarter.`

Chatter Bot : `From this information, it seems the Personal Trading Account plan is best suited to your needs. It comes with a trade account, and charges a commission of 3% per trade.`

Customer : `Thank you.`

– – – – – – – – – – – – – – – End of Conversation – – – – – – – – – – – – – – –

Customer : `Hello. I am unable to access my transactions.`

Chatter Bot : `Can you please state your account number?`

Customer : `It is ACT1296.`

Chatter Bot : `What is your mothers maiden name?`

Customer : `It is Rosen.`

Chatter Bot : `Which city were you born in?`

Customer : `Chicago.`

Chatter Bot : `Thank you for verifying your account. Please state the nature of your problem.`

Customer : `I cannot access my transactions.`

Chatter Bot : `What are the transaction IDs?`

Customer : `I do not know them.`

Chatter Bot : `Do you remember what kind of transactions were they?`

Customer : I think there were 3 buy orders and 2 sell orders.

Chatter Bot : Transactions take 48-??72 hours to execute.

Customer : But I need to edit the transactions before they execute.

Chatter Bot : You can edit the transactions from the portfolio menu.

Customer : But I am unable to access them.

Chatter Bot : Are the transactions valid?

Customer : What are valid transactions?

Chatter Bot : Only valid transactions are posted to your account.

Customer: How do I know if the transactions were valid? Why would they be invalid?

Chatter Bot : Invalid transactions are automatically removed from your account.

Customer : Why would my transactions be invalid?

Chatter Bot : `Did you have sufficient margin in your account to execute your buy order?`

Customer : `I think so. Yes.`

Chatter Bot : `Then your transactions were valid.`

Customer: `Then why am I unable to access the transactions?`

Chatter Bot : `You can edit the transactions from the portfolio menu.`

– – – – – – – – – – – – – – – – End of Conversation – – – – – – – – – – – – – – – –

Customer : Hello. I would like to upgrade my account.

Chatter Bot : You can upgrade your account from your homepage. Click on the account settings link and you will get an option to upgrade your account.

Customer : I tried to do that. But I am unable to do it.

Chatter Bot : What is the account you currently have?

Customer : It is the Intermediate Plan 1.

Chatter Bot : And what account are your trying to upgrade to?

Customer : I want to upgrade to Intermediate Plan 2.

Chatter Bot : You need a balance of $5,000 to complete the upgrade. Do you have $5,000 balance in your account to complete the upgrade?

Customer : Yes, I have sufficient balance. I have $6,000.

Chatter Bot : Have you previously upgraded your account in the last 90 days? You are only allowed to make one upgrade every 90 days.

Customer : No. My last upgrade was around 4 months ago.

Chatter Bot : Have you upgraded your account thrice in the last year? You are only allowed 3 upgraded in a year.

Customer : No. I have made only 2 upgrades this year.

Chatter Bot : Did you previously have an account in the Intermediate Plan 2?

Customer : Yes. But I changed it to Intermediate Plan 1 around 9 months ago.

Chatter Bot : It is not possible to change back to a similar account in the same year.

Customer : Oh I see. Thanks.

Chatter Bot : You are welcome.

– – – – – – – – – – – – – – – End of Conversation – – – – – – – – – – – – – – –

Customer : I have a problem with my portfolio.

Chatter Bot : Can you please explain the problem?

Customer : I wanted to change my allocation. Buy $1,000 worth of the call options from the margin in my derivatives.

Chatter Bot : Can you please explain the problem?

Customer : I get an error. It says the transaction is invalid.

Chatter Bot : Do you have sufficient margin in the portfolio?

Customer : Yes. I sold $750 worth of stock in JFD and along with my previous balance of $350, I have sufficient margin in my portfolio.

Chatter Bot : When did you place your sell order?

Customer : Yesterday.

Chatter Bot : What is the current margin being shown in your portfolio?

Customer : It shows $350.

Chatter Bot : `You do not have sufficient margin in the portfolio.`

Customer : `I should have sufficient margin in my portfolio after the execution of the sell order I placed yesterday.`

Chatter Bot : `When did you place your sell order?`

Customer : `Yesterday.`

Chatter Bot : `It takes 48 to 72 hours for a requested transaction to execute.`

Customer : `Ok. I understand now.`

– – – – – – – – – – – – – – – End of Conversation – – – – – – – – – – – – – – –

Customer : `Hello.`

Chatter Bot : `Hello. How can I help you today?`

Customer : `An incorrect commission fee has been charged to my account.`

Chatter Bot : `What is your account number?`

Customer : `It is GSE8593.`

Chatter Bot : `What is your mothers maiden name?`

Customer : `Robin.`

Chatter Bot : `Which city were you born in?`

Customer : `Phoenix.`

Chatter Bot : `What is the transaction ID for which you have a problem?`

Customer : `KLW8740.`

Chatter Bot : `When did you make the transaction?`

Customer : I put in the order on Friday August 19. But I cancelled the order on Wednesday August 24 within the 3-??day period.

Chatter Bot : Transactions usually take 48 to 72 hours to process. Has you transaction been carried out?

Customer : It was carried out.

Chatter Bot : Your transaction went through in 48 hours and hence you have charged the transaction fee.

Customer : Can I do anything to reverse the transaction?

Chatter Bot : Since the transaction is a market transaction, it cannot be reversed.

Customer : Well thanks for nothing then.

Chatter Bot : You are welcome.

– – – – – – – – – – – – – – End of Conversation – – – – – – – – – – – – – –

Appendix B

Goal-Fulfillment Maps

Examples of goal-fulfillment maps.

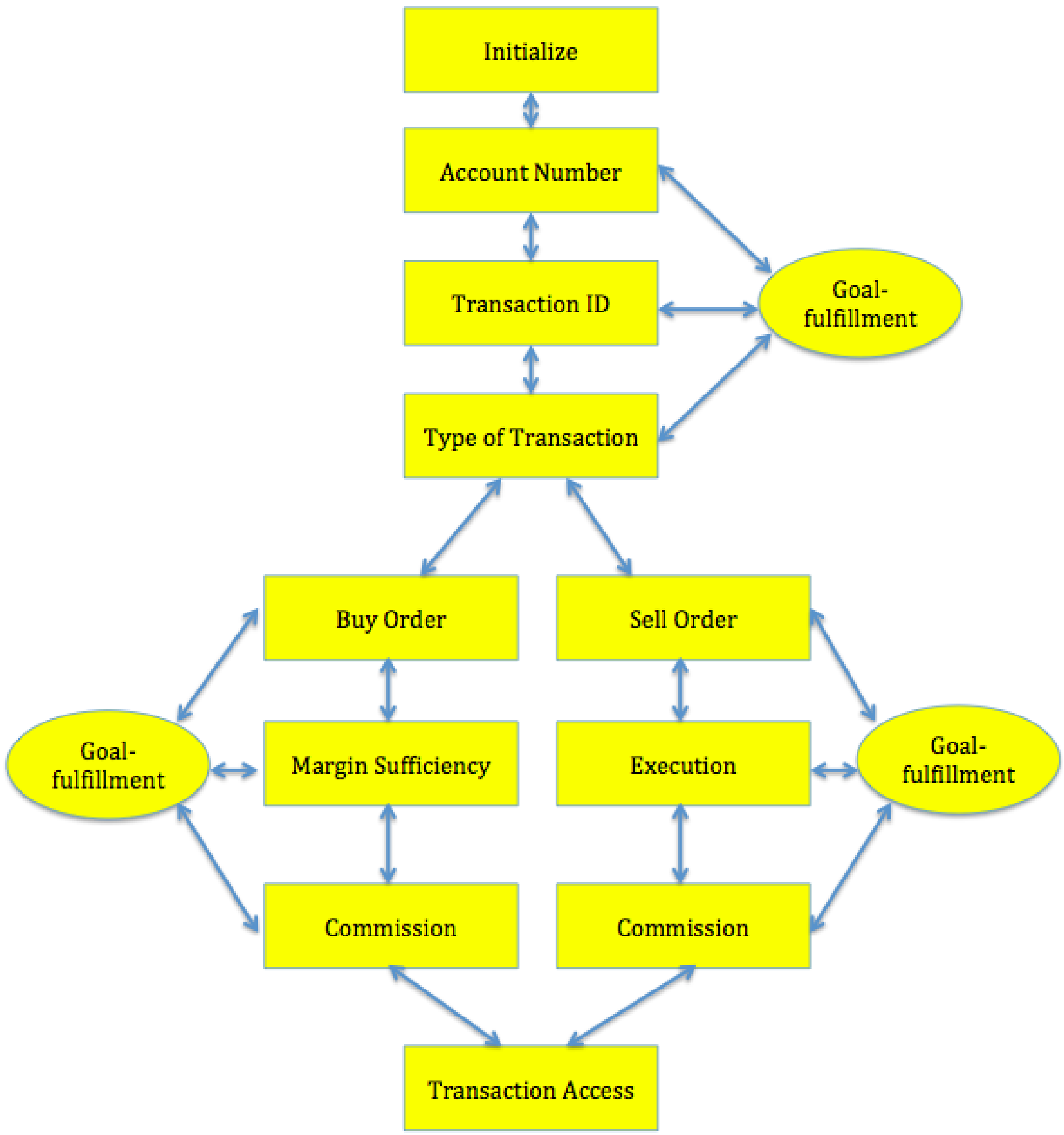

Figure B.1: Goal-fulfillment map [58, 57, 60] of the procedure to verify if all the conditions of a buy or sell order have been met..

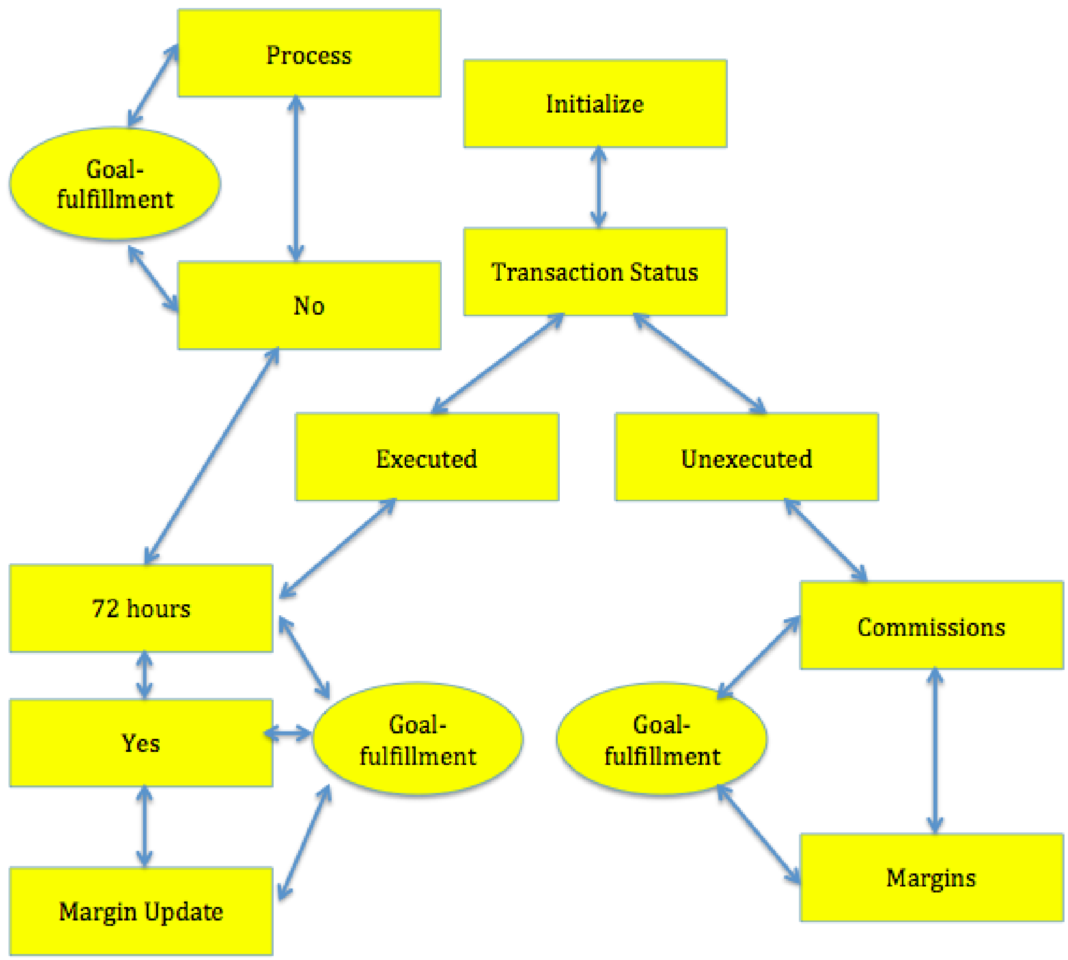

Figure B.2: Goal-fulfillment map [58, 57, 60] of the procedure to process transactions.