Artificial Conversations for Customer Service Chatter Bots: Architecture, Algorithms, and Evaluation Metrics

Chayan Chakrabarti[a,*], George F. Luger[a]

[a]*Computer Science Department MSC01 1130, 1 University of New Mexico, Albuquerque, New Mexico, 87131 USA*

Abstract

Chatter bots are software programs that engage in artificial conversations through a text-based input medium. They are extensively deployed in customer service applications. Existing approaches to artificial conversation generation emphasize grammatical and linguistic modeling techniques. They focus on generation of discrete sentence-level utterances. These approaches perform poorly in conversational situations with contextual continuity over a series of utterances. Natural conversations are processes that adhere to well-defined conventional semantics and are contextually grounded in domain-specific knowledge. We present an approach that combines pragmatic semantics with content semantics to generate artificial conversations in the customer service domain. We also present a specific set of evaluation criteria for evaluating the quality of artificial conversations in the customer service domain. We compare bot generated artificial conversations with human generated natural conversations in this domain. Our evaluation criteria include both subjective and objective metrics. We use Grice's maxims front the theory of pragmatics for some of our metrics. Our subjective metrics are evaluated by a panel of judges. We also present an analysis of the relationships between the metrics and the quality of artificial conversations.

Keywords: Artificial Conversations, Computational Conversations,

*Corresponding Author

Email addresses: `cc@cs.unm.edu` (Chayan Chakrabarti), `luger@cs.unm.edu` (George F. Luger)

Conversation Engineering, Conversation Semantics, Applied Computational Linguistics, Applied Natural Language Processing.

1. Introduction

Many businesses have capitalized on the growth of text-based chat as a medium of communication and implemented their customer support operations through chat interfaces. Existing and potential customers chat with customer service representatives and resolve their customer service issues or seek information about the business' products and services. Analysts predict that by 2015, at least 50% of customer service will be performed by chatter bots, resulting in cost reductions of up to 20% and increased loyalty (Gartner, 2012).

One of the earliest conversational architectures was the GUS (Genial Understander System) (Bobrow et al., 1977), a virtual agent helping a customer make reservations. While the system worked well on handling airline 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 (Seneff et al., 1998; Polifroni and Seneff, 2000) is a client-server architecture for communicating online information like weather and flight information, and consists of 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 (Polifroni and Seneff, 2000). It can handle a very sophisticated range of conversations ranging from yes-no questions to answering complex queries (Filisko and Seneff, 2003). But it is not set up to build the knowledge base using facts, only in terms of anticipated questions (Filisko and Seneff, 2003).

The DARPA Communicator project (Levin et al., 2000) was an initiative to support advanced conversational capabilities including negotiation, plan optimization, and complex explanations. 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 (Walker et al., 2000).

The *Bayesian Receptionist* at Microsoft Inc (Horvitz and Paek, 2000), employed a set of Bayesian user models to interpret the goals of speakers given evidence gleaned from a natural language parse of their utterances. Multiple levels also allowed for the establishment of common ground (Clark, 1996) about uncertainties at each level. Paek and Horovitz (Paek and Horvitz, 2000) then demonstrated how conversations could be modeled as an inference and decision making problem under uncertainty.

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 (Deng et al., 2013). Some specialized techniques leverage dialogue structure in specific context to improve accuracy by encoding speech recognition patterns (Metallinou et al., 2013). Neural networks have also been used for deep-learning solutions to this problem (Henderson et al., 2013). 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 (Gasic et al., 2013). Reinforcement learning techniques have also been used for this problem. (Rieser and Lemon, 2013)

Modern chatter bots implementations can effectively leverage computational linguistics techniques like semantic parsing (O'Shea et al., 2009b) and sentiment analysis (Whitehead and Cavedon, 2010). Contemporary chatter bots perform very effectively in question-answer settings and other similar utterance-exchange pair settings, where the context of the conversation are independent from one exchange to the next (Mauldin, 1994; Saygin and Ciceklib, 2002; Chakrabarti and Luger, 2012). However, they perform poorly in conversational situations where a specific context is maintained through a series of several utterance-exchange pairs. Existing customer service chatter bots are able to handle FAQ-type queries, but are unable to handle contexts that require a short conversation (Chakrabarti and Luger, 2012; Chakrabarti, 2014). Also, most chatter bot implementations 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. They do not make an explicit distinction to modeling the content required for the conversation and the semantics inherent in the conversation process.

We present models for generating text chat-based artificial conversations by combining content semantics with pragmatic semantics. Our conversations are restricted to the customer service domain. We learn the parameters of our models from a corpus of customer service conversation logs of an online electronic trading portal.

2. Corpus of Conversations

We used a corpus of chat transcripts between a human customer and a human customer service agent working for an online electronic trading portal. 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. An utterance is everything that is said by either the customer or the representative in a single turn. It consists of one for more sentences. We assume that each utterance belongs to a single context.

A series of successive utterance pairs on the same context constitutes a conversation. The shortest conversation had 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 representative. Most of the conversations were related to single context. The conversations that were not interleaved and related to more than one context were not analyzed.

3. Models for Artificial Conversations

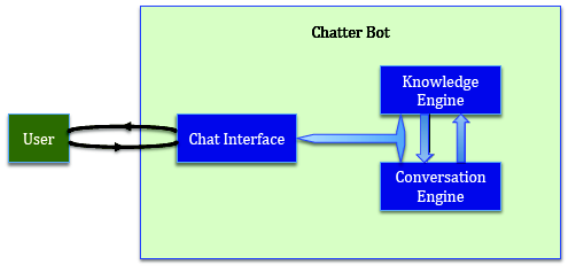

Figure 1: Overview of architecture

We generate artificial conversations through the interaction of pragmatic semantics, implemented by a conversation engine, and content semantics, implemented by a knowledge engine. Essentially, the knowledge engine models the *what to say* aspect of the conversations, and the conversation engine models the *how to say it* aspect of the conversation. A separate chat interface module interacts directly with the human and performs lower level pre-processing tasks as shown in Figure 1.

3.1. Content Semantics Modelling

The knowledge engine models the content semantics of the conversation. The two main content defining characteristics are the domain-specific information about the subject matter being discussed and the particular speech act being adhered to. We use the following speech acts identified from the corpus. The first 5 are from Searle's illocutionary speech act (Searle, 1969, 1975), while the last one is a dialog act (Stolcke et al., 2000) defined for this application.

3.1.1. Assertive Speech Act

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.

3.1.2. *Directive Speech Act*

Utterance serves as a request, command, advice, or instruction.

3.1.3. *Commissive Speech Act*

Utterance 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 and all other participants commit to it's veracity.

3.1.4. *Expressive Speech Act*

Utterance expresses some emotion or attitude about the context of the conversetion.

3.1.5. *Declarative Speech Act*

Utterance causes the state of the world to change, with full implicit or explicit acknowledgement of the participants.

3.1.6. *Goal-Fulfillment Speech Act*

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.

The speech act is selected using a bag-of-words based latent semantic approach (Section 4.2.)

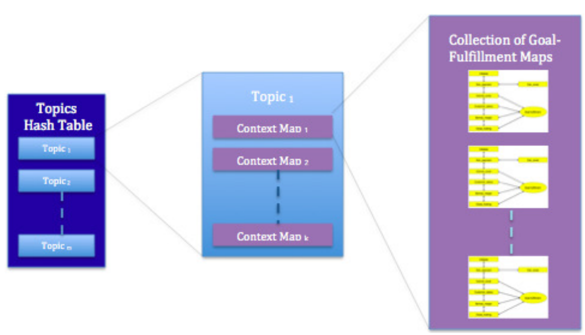

Figure 2: The Topics Hash Table is a collection of contexts modeled by goal-fulfillment maps and identified by a two-valued hash key: the topic and the type of conversations.

A Topics Hash Table (Figure 2.) organizes the set of topics within the scope of the conversation using a two-valued hash key. One hash key is the topic. We have modeled conversations pertaining to 9 topics under 3 broad umbrellas: Accounts (Login, Configuration, Open), Balance (Margins, Transfers, Portfolio), and Transactions (Commissions, Orders, Processing). The relevant topic is selected using a bag-of-words latent semantic approach described in section 4.2. The second hash key identifies the specific context within the topic, corresponding to 4 distinct conversations types as described in section 3.2. These contexts within the topics are modeled using goal-fulfillment maps.

Goal-fulfillment maps are based on the conversational semantic framework introduced by O'Shea (O'Shea et al., 2008, 2009b,a, 2010). They are data structures that seek responses to specific queries (or sub-contexts), which require specific responses in order to make the conversation progress towards a goal. While engaging in a dialogue with the user, the chatter bot captures specific pieces of information from the conversation to progress along the network of contexts described by the goal-fulfillment map (Chakrabarti and Luger, 2012). A context, uniquely identified by the topic and the type of conversation, is implemented as a list of goal-fulfillment maps.

In the example in Figure 3, where a chatter bot advises a customer of an electronic trading website about login issues, the contexts along the goal-fulfillment map expresses specific queries, which require specific answers in order for progression to be made along the designated route. Dialogue will traverse the goal-fulfillment map in a progression starting with the base context named *Initialize*. It is possible to revert to a previously visited context in the case of a misinterpreted line of input. The user can alert the chatter bot that there has been a misunderstanding. For example in following context, *Non Payment* aims to elicit the reason for non-payment of the margin fees; *Can Cover* identifies that the customer does have enough margin and thus goal-fulfillment is achieved; *Cannot Cover* aims to elicit why the customer doesn't have sufficient margin; *Customer Status* identifies the status of the customer, and keeps following the map until goal-fulfillment is achieved.

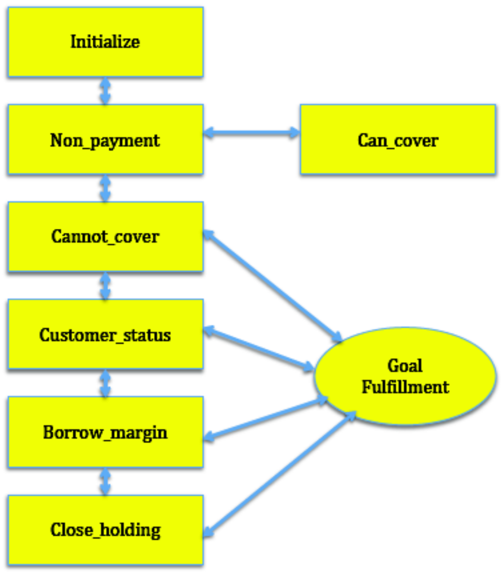

Figure 3: A Goal Fulfillment Map encodes the sequence of questions to be asked, whose responses will enable fulfill a specific goal during the conversation.

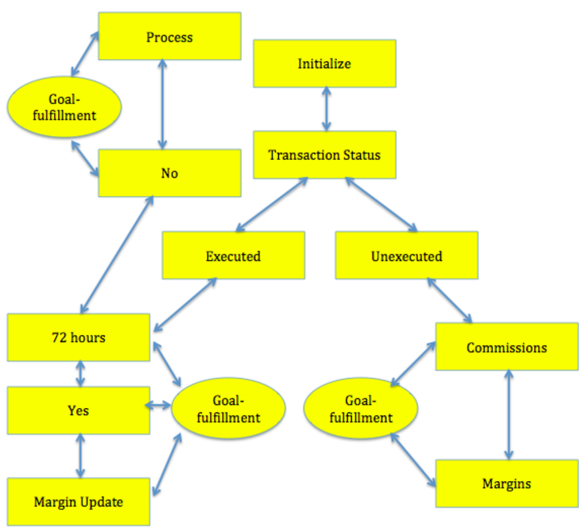

Figure 4: Goal-fulfillment map of the procedure to process transactions.

Figure 4 describes a goal-fulfillment map modeling the procedure to process transactions. It consists of several possible flows of conversations depending

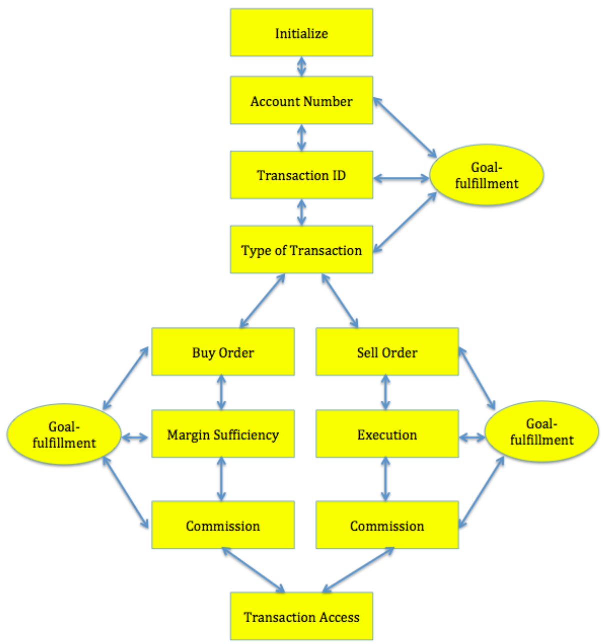

Figure 5: Goal-fulfillment map of the procedure to verify if all the conditions of a buy or sell order have been met.

on the responses obtained and several possible goals can be fulfilled. Similarly, Figure 5 shows a goal-fulfillment map modeling the procedure to verify if all the conditions of a buy or sell order have been met. Again, depending on the specific responses obtained the conversation can proceed in several possible directions.

3.2. Pragmatic Semantics Modelling

The conversation engine models the pragmatics semantics of the conversation. We model the pragmatic semantics of the conversation using a probabilistic finite state automaton (FSA), 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. We have modeled 4 distinct types of conversations using 4 corresponding FSAs: Procedural, Informational, Troubleshooting, and Dispute-Resolution.

The conversations states have been manually identified from the corpus and the transition probabilities are learned from the corpus. The FSAs are conversational grammars, which define the production rules for the particular type of conversation. This is distinct from regular language grammars that generate individual sentences. 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.

3.2.1. Procedural Conversation

(Figure 6.) guides the user through a series of steps to achieve some objective, e.g., change password. Conversation begins in the **Start** state. Small talk like exchange of names and pleasantries happens in the **Greeting** state. In the **Advisory** state, the user is instructed though a series of steps to be followed. Since this is essentially changing the state of the world through actions, utterances of the *Assertive* speech act keeps the conversation in the same state. **Satisfaction** state indicates that the procedure has been completed as verified by utterances of the *Expressive* speech act. An *Assertive* speech act can take

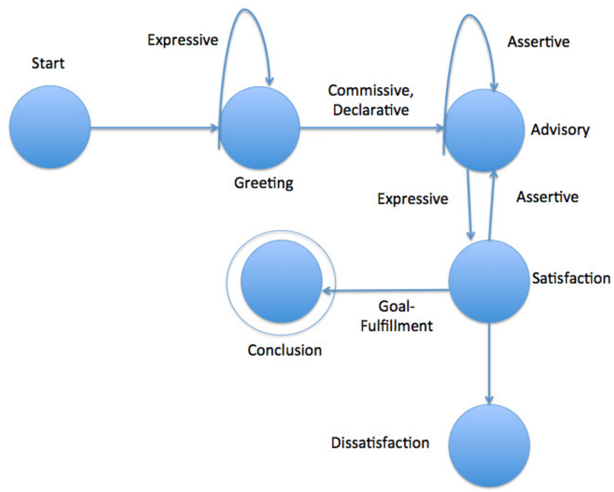

Figure 6: The probabilistic FSA for Procedural Conversations.

it back to the **Advisory** state. **Dissatisfaction** is a dead-end state when the conversation has gone beyond programmatic limits. This indicates conversation failure. **Conclusion** state indicates the end of the conversation process when the last goal-fulfillment task as indicated by the corresponding Topic Hash Table has been achieved.

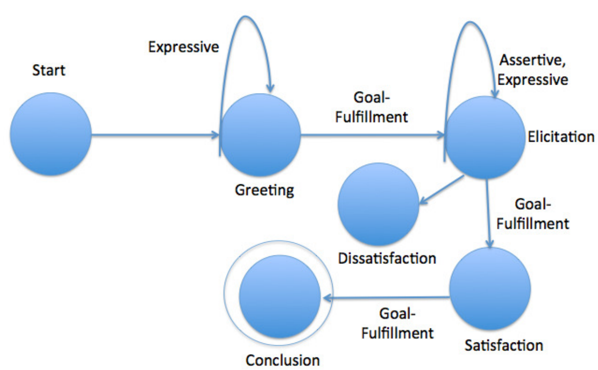

Figure 7: The probabilistic FSA for Informational Conversations.

3.2.2. Informational Conversation

(Figure 7.) provides the user with a set of facts, e.g., cost of a plan. The **Start**, **Greeting**, **Satisfaction**, **Dissatisfaction**, and **Conclusion** states are the same as described before. In the **Elicitation** state, truths or facts about the state of the world are uttered from the relevant situational context encoded by the correspond goal-fulfillment map in the Topic Hash Table. Utterances of 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.

Notice the difference between a procedural and informational conversation. In the former, utterances of *Expressive* speech act cause the conversation to leave the **Advisory** state. In the latter, utterances of the *Expressive* speech act cause 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.

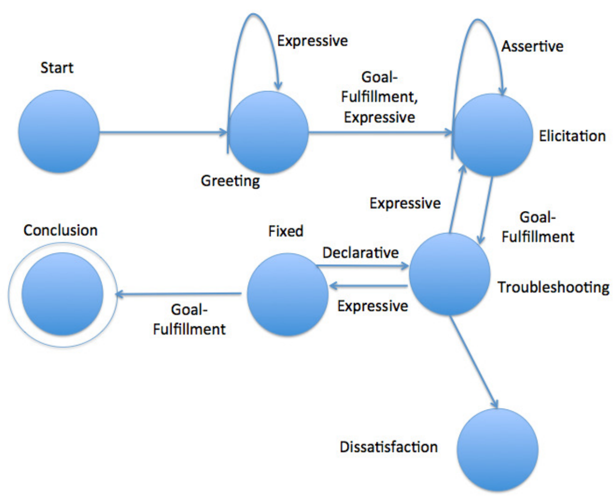

Figure 8: The probabilistic FSA for Troubleshooting Conversations.

3.2.3. Troubleshooting Conversation

(Figure 8) solves a user problem by understanding the nature of the problem and then taking steps to overcome it or seeking more information about it, e.g., transactions not being shown in account. The **Start**, **Greeting**, **Elicitation**, **Dissatisfaction**, and **Conclusion** states are the same as described before. The **Troubleshooting** state tells the user to take steps to change the state of the world in an attempt to resolve the problem as defined by the corresponding goal-fulfillment map in the Topic Hash Table. Note that alternative utterances of the *Declarative* and *Expressive* speech acts 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. The **Fixed** state indicates that the problem issues has been partially or completely resolved. A partial fix would trigger an *Expressive* speech act utterance from the customer taking 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 takes the conversation to the **Conclusion** state.

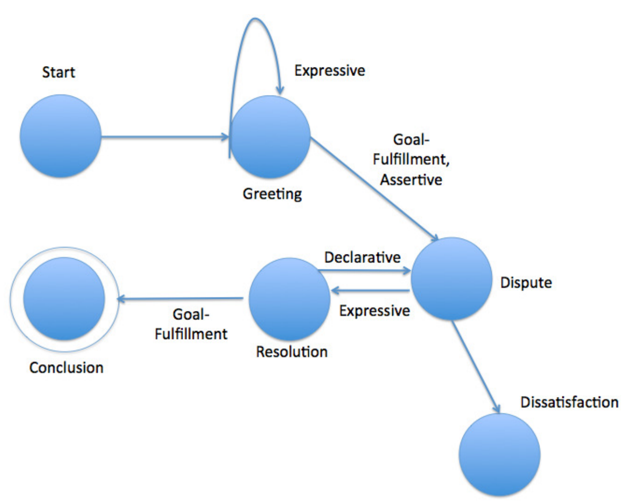

Figure 9: The probabilistic FSA for Dispute Resolution conversations.

3.2.4. Dispute-Resolution Conversation

(Figure 9) resolves a disagreement with user, e.g., incorrect commission being charged. The **Start**, **Greeting**, **Dissatisfaction**, and **Conclusion** states are the same as described before. The **Dispute** state tells the user to take steps to change some truth about the state of the world, or empathize or criticize some aspect of the customer situation or statement as defined 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. The **Resolution** state indicates the problem has been partially or completely fixed. A partial fix would trigger a *Declarative* speech act utterance taking the conversation back to the **Dispute** state. A *Goal-fulfillment* speech act as indicated by the goal-fulfillment map in the corresponding Topic Hash Table takes the conversation to the **Conclusion** state.

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.

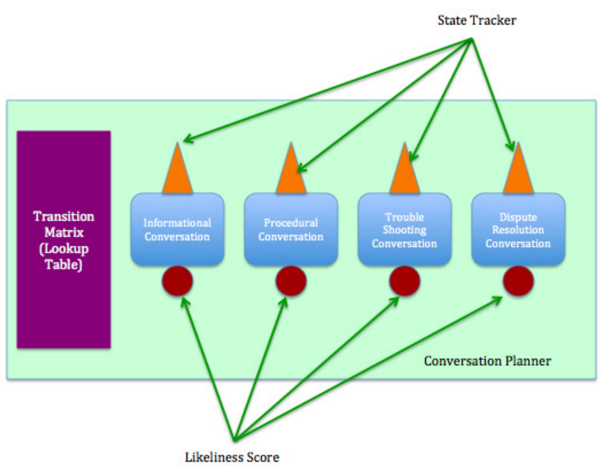

Figure 10: The Conversation Planner consists of the transition matrix lookup table, the state tracker, the likeliness score variable, and the conversation solutions in the workspace.

A conversation planner serves as the workspace for the conversation generation (Figure 10.). It decides which transition to perform in the probabilistic FSA, maintains the likeliness score for each type of conversation, and a counter to keep track of the state for the conversation. It maintains 4 simultaneous solutions corresponding to the 4 conversations types. With each utterance in the conversation, each conversation solution is updated. When a solution becomes highly likely as indicated by a heuristic score described below, it is maintained and all others removed from the workspace. Conversely, when a solution becomes highly unlikely according the heuristic score described below, it is removed from the workspace. Ocassionally all solutions have their heuristic scores fall below the pre-defined threshold and are dropped from the workspace leading to conversation failure. The conversation planner has access to a lookup table of transition probabilities for the 4 probabilistic FSA. The probability values are learned from the corpus. The lookup table indicates which transition to make for each automata. For each conversation solution active in the workspace, a separate tracker is maintained to indicate the current state of the conversation according to that solution. There can be 0 (conversation failure) to 4 state trackers present simultaneously.

For each solution, a likeliness score is maintained indicating how likely it is that this solution is the right one for the specific conversation being generated. The score is an integer variable initialized to 0. When the conversation follows the most likely path indicated by the FSA 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 FSA, 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 FSA and hence that solution is unlikely. When a score becomes an nth multiple of the next highest score, for $n > 8$, where n indicates the number of turns of utterances, then all other solutions are dropped from the workspace. This is an indication that one of the solutions is overwhelmingly likely as compared to the others and should be the only one considered. We selected the value of n and the heuristic values 0 and 3 by trial and error, and repeated tweaking. Such a solution was implemented because there is some evidence that human beings process conversations in this way (Changeux, 1998; Craig and Tracy, 1983). Literature from linguistic neurobiology suggests that human beings maintain several alternative solutions while processing conversations (Clarke, 1983; Winograd and Flores, 1986)

4. Learning Model Parameters

We learn the topics and the speech acts from the corpus described in section 2.

4.1. Topics and goal-fulfillment maps

We manually tagged each conversation in the corpus with one of the 4 conversation types. For each of the 9 topics, specific contexts and the corresponding conversation types were manually identified. These included: 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), and 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 from the corpus, the specific domain knowledge for the 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. Then the goal-fulfillment maps for all combination of contexts and conversation types were manually created and encoded in the Topic Hash Table.

4.2. Speech Acts

We used a bag-of-words based latent-semantic algorithm to tag each utterance in each conversation in the corpus with a speech act. In addition to the standard bag-of-words list defined in lexical taxonomy of speech and dialogue acts (Moldovan et al., 2011) we also added the following words:

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

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

3. Commissives {"will cancel account", "will reactivate account", "will deactivate account", "orders will be reinstated", "margin will be restored", "commis-

sions will be removed”}

4. Expressives {”glad”, ”happy”, ”upset”, ”unhappy”, ”unacceptable, ”acceptable” }

5. Declaratives {”account closed”, , ”account reactivated”, ”account deactivated”, ”configuration changed”, ”orders cancelled”, ”orders reinstated”, ”margin restored”}

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.

4.3. Topics

We used a bag-of-words based latent-semantic algorithm to tag each conversation in the corpus with one of 9 topics. The following bag-of-words was used.

1. Login { ”Login”, ”Password”}

2. Configuration {”Configuration”, ”Upgrade”, ”Daily”, ”Regular” }

3. Access {”Access” }

4. Margins {”Margins”, ”Balance” }

5. Transfers {”Transfers”, ”Allocation” }

6. Portfolio {”Portfolio”, ”Commodity”, ”Equity”, ”Trade” }

7. Commissions: {”Commissions”, ”Charge”, ”Cost” }

8. Orders {”Orders”, ”Buy”, ”Sell” }

9. Processing {”Processing”, ”Reinstate”, ”Cancel”, ”Execute” }

5. Artificial Conversation Generation

We generate the conversation by interacting with the chatter bot architecture via a standard terminal. We play the role of the customer of the online electronic trading website one of the issues mentioned in section 4.1. and typing it out on the standard terminal. The chat interface performs pre-processing of the input, like autocorrecting for spelling and grammar, and stemming(Porter, 1980).

The knowledge engine uses the stems as input for the latent semantic algorithm to select the topic and the speech act. The conversation engine initially maintains solutions for all 4 types of conversation. The specific goal-fulfillment map is picked out of the Topic Hash Table using the conversation context and the topic as the keys. The goal-fulfillment makes the bot make an utterance eliciting specific responses.

The response of the bot is output to the standard terminal through the chat interface. As the conversation progresses through utterance exchanges, the states are advanced in the FSAa corresponding to the conversation solutions in the workspace. Eventually, either the conversation fails due to all solutions being dropped from the workspace, or a *Dissastisfaction* state reached in FSAs of active solutions, or the conversation successfully concludes when a single active solution remains in the workspace and an accepting *Conclusion* state is reached in the corresponding FSA.

5.1. Anatomy of a conversation

We show an example of how a conversation is generated through a single utterance exchange pair. The conversation starts with a human making a comment (Chakrabarti and Luger, 2014).

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

The chat interface identifies the stems: "account", "day trade", "open", and "options". The knowledge engine uses latent semantic analysis to determine that the type of speech act is *Expressive*, since the bag-of-words included "would" and "like" and that the topic is *Open* since the bag-of-words included "new", "account", and "open". In the conversation engine, initially all four possible solutions are maintained. A counter is initialized to maintain the current state of the conversation in each solution. The appropriate goal-fulfillment map is then pulled out using the topic key and context key. The map encodes the steps for opening a new account. The first step in the map makes the bot make the following utterance.

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

This process is repeated until the end of the conversation is indicated by the conversation planner counter being in an accepting state (Chakrabarti and Luger, 2014).

6. Evaluation of Artificial Conversations

How do we differentiate between a satisfactory and unsatisfactory conversation? It is important to have a standard, consistent metric to measure the quality of artificial conversations as a means for comparison between different approaches, and a benchmark for seeking improvement. In this paper, we define some metrics to evaluate artificial conversations.

We then demonstrate how these metrics can be used to evaluate artificial conversations from the customer service domain. For testing, we had access to a corpus of actual human conversations between a customer and a customer service agent working for an online electronic trading company. We also had access to a set of 48 artificial conversations generated by a semantically aware conversation bot in the same domain that was trained using conversation modeling parameters learned from the corpus

6.1. 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 (Mey, 2001). 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 (Grice, 1957, 1975, 1989; Mey, 2001).

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.

6.2. 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 (Grice, 1957, 1975, 1989).

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. (Saygin and Ciceklib, 2002) 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.

6.3. Situation Specific Conversations

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 e.g., 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.

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. Also, since it was emphasized that semantically aware chatter bots can got beyond single utterance-response exchange pairs, it follows that scoring the number of utterance-exchange pairs over which the artificial conversation can maintain coherence is an important evaluation metric.

6.4. Objective and Subjective Metrics

We show how Grice's Maxims can be used to evaluate artificial conversations. 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.

6.4.1. Quality Maxim.

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.

6.4.2. Quantity Maxim.

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.

6.4.3. Relation Maxim.

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.

6.4.4. Manner Maxim.

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 Section 3. for specifically evaluating customer service conversations. These are described below.

6.4.5. 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. 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.

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.`

Assume that the domain specifies the following followup questions for this specific situation.

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

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

3. 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%.

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

$$\frac{Number\ of\ successful\ conversations}{Total\ number\ of\ conversations}$$

6.4.7. Number of Coherent Conversation Turns:

In Section. 1, it was shown that semantically aware conversation bots can go beyond a single utterance-response exchange and hold short conversations where the context is maintained across several such pairs. Hence, an important metric is to determine the number of utterance-exchange pairs over which the chatter bot is coherent.

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. Experimental Setup

We selected 16 natural conversations from the corpus described in Section 1., and we had a set of 48 artificial conversations generated by a semantically aware chatter bot (Chakrabarti and Luger, 2013). The set of 64 conversations were then divided into 8 different subsets. Each subset consisted of 2 natural conversations and 6 artificial conversations. The transcripts of these conversations can be found here `www.cs.unm.edu/~cc/artificial_conversations/transcripts/`

We selected a panel of human judges consisting of 48 freshman students in an introductory Computer Science class for non-majors. Each judge was given a subset of the conversations, asked to read the conversation transcripts, and then grade each conversation on a 0-5 Likert scale if they agreed that the conversation satisfied the Quantity, Manner, and Relation maxims, with 0 being the worst score, and 5 being the best score. The judges weren't told which conversations were natural and artificial. Thus each conversation, natural and artificial, was graded by 6 judges.

8. Results and Discussion

The grades from the entire panel of judges for all the subsets of conversation transcripts were collected. Simple statistical analysis was performed on them. As mentioned in the Section 5., there were 8 distinct subsets of conversations. Each subset had 2 natural conversations and 6 artificial conversation and the judges weren't told which was which.

The raw score given by each judge was on a 0 to 5 continuous Likert scale. The scores for the natural conversations given by each judge was used to normalize that judges' score for the artificial conversations for each maxim. For example, if a judge assigned scores of 4.3 and 4.1 for the Manner Maxim for the 2 natural conversations, 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 all the artificial conversations, then these scores would be normalized by the average score for the natural conversations. Thus the scores for this judge for the artificial conversations 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.

The scores for the objective metrics, Quality Maxim, Percentage of Follow up, and Number of Coherent Turns were calculated by us. Table 1. shows the summary statistics of all these scores.

Table 1: Summary statistics for all the quantifiable metrics.

Metric	**Avg.**	**St. Dev.**	**T-test**	**Significant?**
Quality	0.80	0.18	$p < 0.0001$	extremely
Quantity	1.05	0.13	$p < 0.05$	yes
Relation	0.85	0.10	$p < 0.0001$	extremely
Manner	0.85	0.12	$p < 0.0001$	extremely
% Follow Up	0.86	0.21	$p < 0.0001$	extremely
# Turns	5.88	1.12	NA	NA

The averaged normalized scores for artificial conversations for the Manner Maxim and Relation Maxim were lower than 1.00. This means that the judges graded the natural conversations higher than the artificial conversations on these criteria. This means, according to them, the human does better than the bot for these criteria. The average score across all artificial conversations for the Quantity Maxim, Manner Maxim, and Relation Maxim are 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.

Interestingly, the artificial conversations received a higher grading from the judges 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.

The average number of coherent turns of utterance-exchange pairs across all artificial conversations is 5.88. 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.

Overall, 42 out of the 48 conversations were successful in resolving the customer issue for a success rate of 87.5%. 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 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 2. shows that the correlation between the success of the artificial conversation and the Quantity Maxim is fairly low. There are some unsuccessful conversations 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 3. shows that success is fairly correlated with the Relationship Maxim. Figure 7.4 shows that success is highly correlated with the Manner

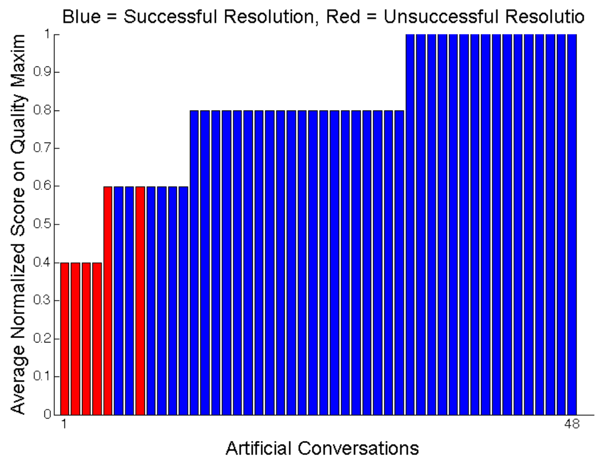

Figure 11: Relationship between successful and unsuccessful resolutions in the artificial conversations and the average normalized score for the Quality Maxim. The x-axis shows each of the 48 artificial conversations, and the y-axis shows the average normalized score. Blue indicates a successful resolution and red indicates an unsuccessful resolution.

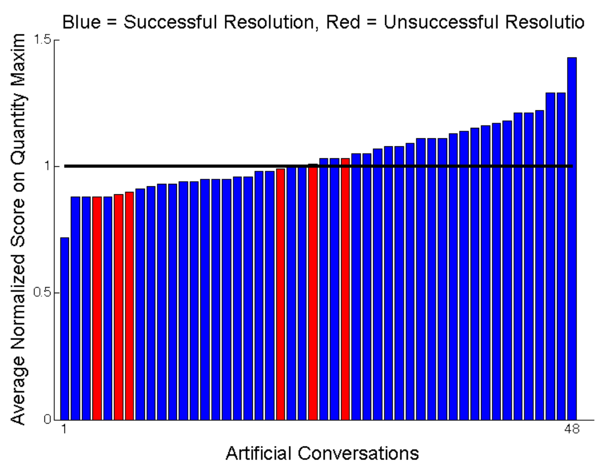

Figure 12: Relationship between successful and unsuccessful resolutions in the artificial conversations and the averaged normalized score for the Quantity Maxim. The x-axis shows each of the 48 artificial conversations, and the y-axis shows the average normalized score. The black horizontal line indicates 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.

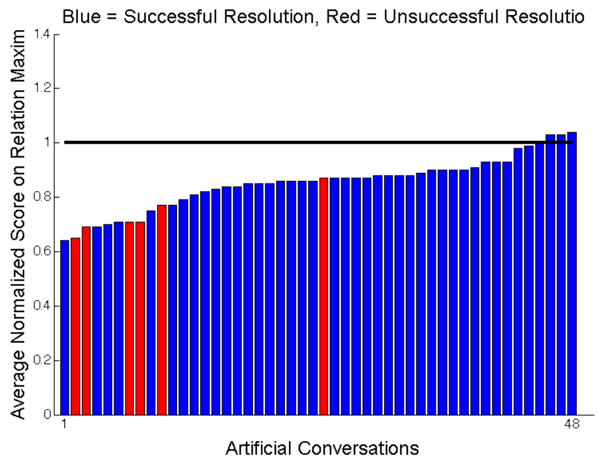

Figure 13: Relationship between successful and unsuccessful resolutions in the artificial conversations and the averaged normalized score for the Relation Maxim. The x-axis shows each of the 48 artificial conversations, and the y-axis shows the average normalized score. The black horizontal line indicates 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.

maxim.

Interestingly, Figure 5. indicates that success is almost 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 6. also shows that success is highly correlated with the number of coherent turns. An artificial conversation with higher number of coherent turns is more likely to successfully resolve the issue.

9. Conclusions

Our chatter bot was able to overcome the limitations described in section 1. It was able to go beyond simple utterance-exchange type conversations like question-answer sessions and hold a short conversation where the context was maintained throughout the conversation. This enabled it to perform reasonably

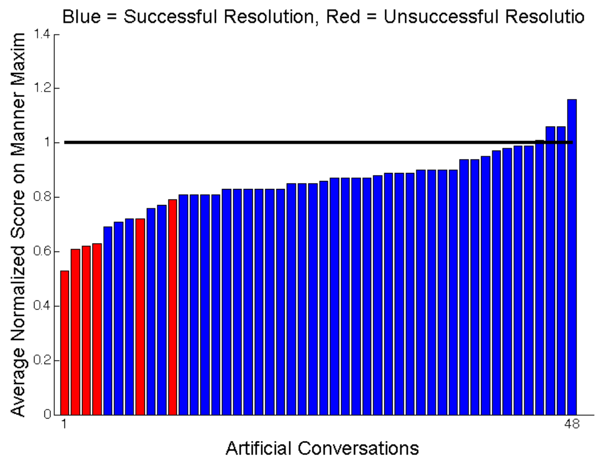

Figure 14: Relationship between successful and unsuccessful resolutions in the artificial conversations and the averaged normalized score for the Manner Maxim. The x-axis shows each of the 48 artificial conversations, and the y-axis shows the average normalized score. The black horizontal line indicates 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.

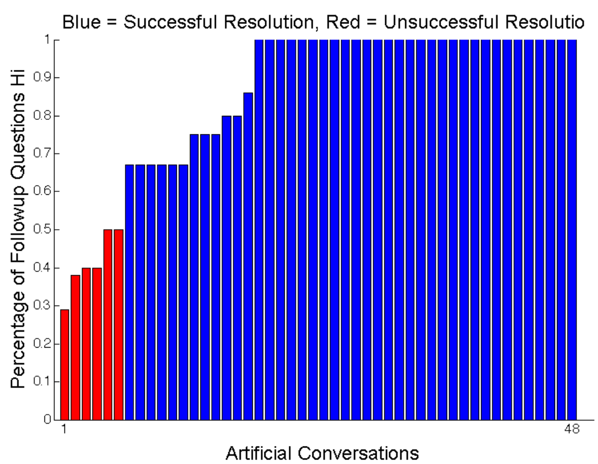

Figure 15: Relationship between successful and unsuccessful resolutions in the artificial conversations and the follow up percentage. The x-axis shows each of the 48 artificial conversations, and the y-axis shows the average normalized score. Blue indicates a successful resolution and red indicates an unsuccessful resolution.

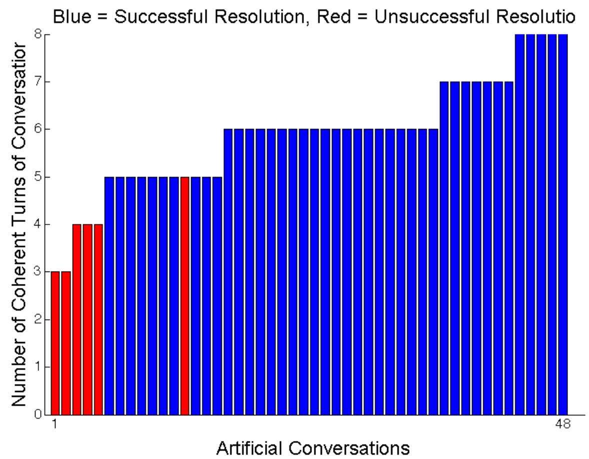

Figure 16: Relationship between successful and unsuccessful resolutions in the artificial conversations and the number of coherent turns.. The x-axis shows each of the 48 artificial conversations, and the y-axis shows the average normalized score. Blue indicates a successful resolution and red indicates an unsuccessful resolution.

effectively in customer service conversations, and handle commonly occurring situations in this domain.

We demonstrated 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.

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. Domain specific situational evaluation metrics were defined suitable for customer service conversations. Natural conversations from the corpus were used as a benchmark to compare the performance of the chatter bot in generating artificial conversations.

Conversation theory and the theory of pragmatics have been well established scientific fields for several decades. It follows that since the our goal is 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. These criteria are grounded in the scientific literature used to evaluate natural conversations by humans.

A key take away is that the Quantity Maxim was be perceived to be less important as compared to the Quality, Relation, and Manner Maxims by a 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 bot design.

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. Also, the process of grading by human judges introduced a feedback lag in the iterative research process. An automated evaluation mechanism that relies on a set of objective metrics could be an important future direction.

Possible future directions include modeling more conversation types, modeling conversation repair to handle failed conversations, and modeling conversations with more than one context.

References

Bobrow, D.G., Kaplan, R.M., Kay, M., Norman, D.A., Thompson, H., Winograd, T., 1977. Gus: A frame-driven dia—og system. Artificial Intelligence

8, 155–173.

Chakrabarti, C., 2014. Artificial Conversations for Chatter Bots Using Knowledge Representation, Learning, and Pragmatics. Ph.D. thesis. University of New Mexico. Albuquerque, NM.

Chakrabarti, C., Luger, G., 2012. A semantic architecture for artificial conversations, in: The 13th International Symposium on Advanced Intelligent Systems, IEEE Press, Kobe, Japan.

Chakrabarti, C., Luger, G., 2013. A framework for simulating and evaluating artificial chatter bot conversations, in: The 26th International Florida Artificial Intelligence Research Society Conference, AAAI Press, St. Pete Beach, FL.

Chakrabarti, C., Luger, G., 2014. An anatomy for artificial conversation generation in the customer service domain, in: 25th Modern Artificial Intelligence and Cognitive Science Conference, pp. 80–85.

Changeux, J.P., 1998. Conversations on Mind, Matter, and Mathematics. Princeton University Press.

Clark, H., 1996. Using language. Cambridge University Press, Cambridge.

Clarke, D., 1983. Language and Action, A Structural Model of Behavior. Pergamon Press.

Craig, R., Tracy, K., 1983. Conversational Coherence. volume 2. Sage Publications.

Deng, L., Li, J., Huang, J.T., Yao, K., Yu, D., Seide, F., Seltzer, M., Zweig, G., He, X., Williams, J., Gong, Y., Acero, A., 2013. Recent advances in deep learning for speech research at microsoft, in: IEEE International Conference on Acoustics, Speech, and Signal Processing.

Filisko, E., Seneff, S., 2003. A context resolution server for the galaxy conversational systems, in: Proc. Eurospeech.

Gartner, 2012. Organizations that integrate communities into customer support can realize cost reductions of up to 50 percent. Gartner Customer 360 Summit 2012 URL: https://www.gartner.com/it/page.jsp?id=1929014.

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

Grice, P., 1957. Meaning. The Philosophical Review 66.

Grice, P., 1975. Logic and conversation. Syntax and Semantics 3, 41–58.

Grice, P., 1989. Studies in the Way of Words. Harvard University Press.

Henderson, M., Thomson, B., Young, S., 2013. Deep neural network approach for the dialog state tracking challenge, in: SigDial Metz FranceMetz France.

Horvitz, E., Paek, T., 2000. A Computational Architecture for Conversation. Technical Report. Microsoft Research.

Levin, E., Narayanan, S., Pieraccini, R., Biatov, K., Bocchieri, E., Fabbrizio, G.D., Eckert, W., Lee, S., Pokrovsky, A., Rahim, M., Ruscitti, P., Walker, M., 2000. The att-darpa communicator mixed-initiative spoken dialog system, in: ICSLP.

Mauldin, M., 1994. Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition, in: Proceedings of the Eleventh National Conference on Artificial Intelligence, AAAI Press, Seattle, Washington.

Metallinou, A., Bohus, D., , Williams, J.D., 2013. Discriminative state tracking for spoken dialog systems, in: Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL), Sofia, Bulgaria.

Mey, J.L., 2001. Pragmatics: An Introduction. 2 ed., Oxford: Blackwell.

Moldovan, C., Rus, V., Graesser, A., 2011. Automated speech act classification for online chat, in: The 22nd Midwest Artificial Intelligence and Cognitive Science Conference.

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

O'Shea, K., Bandar, Z., Crockett, K., 2009a. A semantic- based conversational agent framework, in: The 4th International Conference for Internet Technology and Secured Transactions (ICITST-2009), Technical Co- Sponsored by IEEE UK?RI Communications Chapter, London. pp. 92–99.

O'Shea, K., Bandar, Z., Crockett, K., 2009b. Towards a new generation of conversational agents using sentence similarity. Advances in Electrical Engineering and Computational Science, Lecture Notes in Electrical Engineering 39, 505–514.

O'Shea, K., Bandar, Z., Crockett, K., 2010. A conversational agent framework using semantic analysis. International Journal of Intelligent Computing Research (IJICR) 1.

Paek, T., Horvitz, E., 2000. Conversation as action under uncertainty, in: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI), pp. 455–464.

Polifroni, J., Seneff, S., 2000. Galaxy-ii as an architecture for spoken dialogue evaluation, in: LREC.

Porter, M.F., 1980. An algorithm for suffix stripping. Program 14, 130–137.

Rieser, V., Lemon, O., 2013. Reinforcement Learning for Adaptive Dialogue Systems: A Data-driven Methodology for Dialogue Management and Natural Language Generation. Springer.

Saygin, A.P., Ciceklib, I., 2002. Pragmatics in human-computer conversation. Journal of Pragmatics 34, 227–258.

Searle, J., 1969. Speech Acts. Cambridge University Press.

Searle, J., 1975. Indirect speech acts. Academic Press, New York. chapter 3. Speech Acts, pp. 59–82.

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

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Ess-Dykema, C.V., Meteer, M., 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. Computational Linguistics 26, 339–373.

Walker, M., Y, L.H., Y, J.A., 2000. Evaluation for darpa communicator spoken dialogue systems, in: Second International Conference on Language Resources and Evaluation.

Whitehead, S., Cavedon, L., 2010. Generating shifting sentiment for a conversational agent, in: NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, Association for Computational Linguistics. pp. 89–97.

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