

# A Semantic Architecture For Artificial Conversations

Chayan Chakrabarti  
Department of Computer Science  
University of New Mexico  
Albuquerque, New Mexico 87131  
Email: cc@cs.unm.edu

George F. Luger  
Department of Computer Science  
University of New Mexico  
Albuquerque, New Mexico 87131  
Email: luger@cs.unm.edu

**Abstract**—Artificial conversations have many applications in chatter bot-based customer service including website navigation tools and guided online shopping. Existing approaches to generating conversations leverage linguistic and stochastic principles, where lower level grammatical and structural artifacts are modeled. These approaches perform well in pairwise utterance exchanges, but not so well in longer conversational contexts. We simulate more meaningful chatter bot conversations using an architecture that can leverage content and context. Grice's cooperative maxims, which form the central idea in the theory of pragmatics, is our framework for evaluation. The domain of our research is customer service situations, and we compare our artificial conversations with actual conversations of existing chatter bots deployed in the same domain.

**Index Terms** Artificial Conversations, Conversation Architecture, Conversation Modeling.

## I. INTRODUCTION

Chatting with computer programs has come a long way since the pioneering artificial intelligence demonstrations like ELIZA and Parry. Early chatter bots were designed as testing platforms for ideas in NLP. Current chatter bots are more sophisticated and have found application 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 will result in cost reductions of up to 20% along with increased customer loyalty [1].

Many businesses have implemented chatter bots to simulate human customer service representatives. Customers can now chat directly with the chatter bot, without any human intervention. 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. Chatter bots are more reliable, make less mistakes, can be easily scaled to work larger loads, can be modified without extensive retraining, and many instances can work in parallel.

However contemporary customer service chatter bots suffer from one glaring drawback. We illustrate this with an example. 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 my password. But it requires me to enter my Customer Relationship Number. I do not have this number.

*ChatterBot:* 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.

Conversation Transcript 1: Conversation between chatter bot and a customer trying to resolve a login issue.

The chatter bot did not do well with the conversation. If we observe the conversation closely, we notice a definite pattern. When the bot's response is evaluated only in relation to the immediate previous utterance by the human, it grades satisfactorily. It is only when evaluated on a longer, sustained conversation, that it grades poorly. It performs adequately in an isolated question-answer exchange. It even does well 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.

In this research, our aim is to improve the conversational power of chatter bots. Instead of just being able to engage in question-answer exchanges, we want to design agents that are able to hold a longer conversation and more closely emulate the behavior of a human representative. We extend the language model using a conversation architecture that goes beyond grammatical and linguistic details and models the conversational aspects of language.

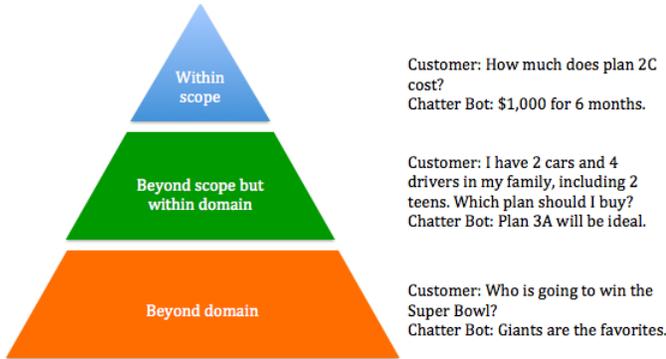


Fig. 1. Classes of Conversations: *Within scope* denoting specific conversations within the domain, *Beyond scope* denoting non-specific open-ended conversations within the domain, and *Beyond domain* denoting conversations beyond the domain being considered.

Chatter bots can carry out conversations ranging from open ended general chitchat to specific questions-answer sessions. To achieve measurable success in our goal, we restrict the domain of conversations to customer service contexts in which a customer calls up a business and engages the chatter bot to seek resolution for customer service problems. Even in this domain, the range of conversations is too large for scientific analysis. Our analysis of customer service chat logs leads us to assert that there is a pyramid-type distribution to conversations (Figure 1). The classes of conversations in the bottom tier can include anything in the whole space of conversations. 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 our analysis we restrict the space of conversations to the top two tiers of this pyramid.

## II. KNOWLEDGE REPRESENTATION FOR CONVERSATIONS

Text-based chatter bot scripts are typically organized into contexts consisting of a number of hierarchically organized rules. Each rule possesses a list of structural patterns of sentences and an associated response [2]. An inordinate amount of scripting is required in order to anticipate the many different ways a user may send an input string [3]. Potentially large numbers of rules and patterns are required. This is a time-consuming and high maintenance task. Also, modifying one rule or introducing a new rule into the script invariably has an

impact on the remaining rules [4]. A reassessment of the entire script would be warranted, without which, the chatter bot is rendered futile. The scripter is therefore required to remember the rankings of the rules and predict how the introduction of new rules will interact with existing rules [5]. It has been shown that by employing sentence similarity measures, scripting can be reduced to a few prototype sentences [6].

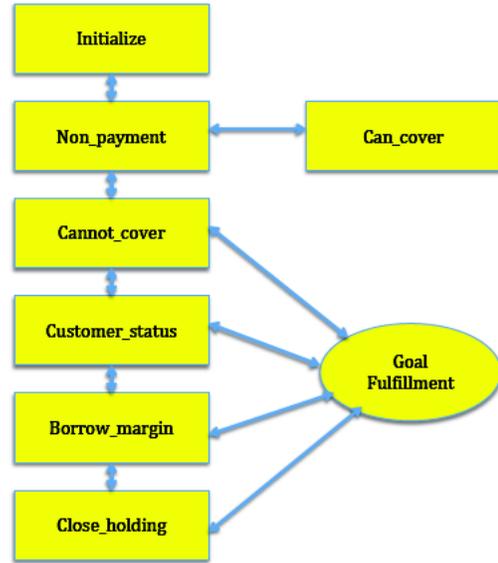


Fig. 2. A Goal Fulfillment Map: where a specific map encodes the sequence of questions to be asked, whose responses will enable fulfill a specific goal during the conversation.

A goal-fulfillment map is an effective way to represent the condensed knowledge in scripts. It is based on the conversation agent semantic framework proposed by O'Shea [7]. The structure of the scripts is described using a new technique known as a goal-fulfillment pathway structured to work in a goal-oriented manner. The contexts are arranged in an order that a bot must traverse in order to reach goal fulfillment. The bot may switch between contexts, traversing forwards or backwards along the goal-fulfillment map. 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. Engaging in dialogue with a user, the chatter bot is able to capture specific pieces of information to progress along the network of contexts described by the goal-fulfillment map [8].

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. Rules that pose a question possess a series of successor rules which seek to obtain an appropriate answer. If no relevant input is sent, a default rule is returned prompting the user for further input to satisfy the request. Each rule is assigned an activation level of one. Once a rule has fired its activation level is reduced to zero and will not fire again from subsequent user input. If a rule has performed its task of eliciting a piece of information from the user, its purpose

is no longer required.

The example domain (Figure 2) is concerned with advising customers of an electronic trading service. The goal-fulfillment map contains several 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. 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.

### III. CHATTER BOT ARCHITECTURE

The key aspects for holding a conversation are knowing what to say that is both relevant to the context and within the domain. The architecture for achieving this is shown in Figure 3. The knowledge engine keeps the conversation in the right domain, while the conversation engine keeps it relevant to the context. These two modules handle distinct tasks in the conversation process. The chat interface module is the first, direct layer of interface with the user.

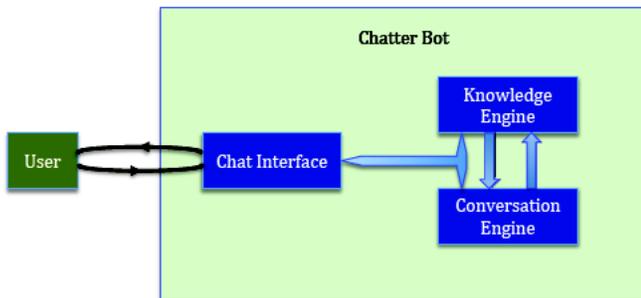


Fig. 3. System Architecture for the Chatter Bot containing the Chat Interface, which pre-processes the raw chat text, the Knowledge Engine, which provides the content of the conversation, and the Conversation Engine, which manages the semantic context of the conversation. The block arrows indicate direction of flow of information between the interfaces implementing the modules.

#### A. Chat Interface

The Chat Interface (Figure 4) 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 Utterance Bucket is an interface that receives the chat text from the user, and places the text into a buffer. It has a spell checker functionality, so that typos can be corrected before putting the text in the buffer.

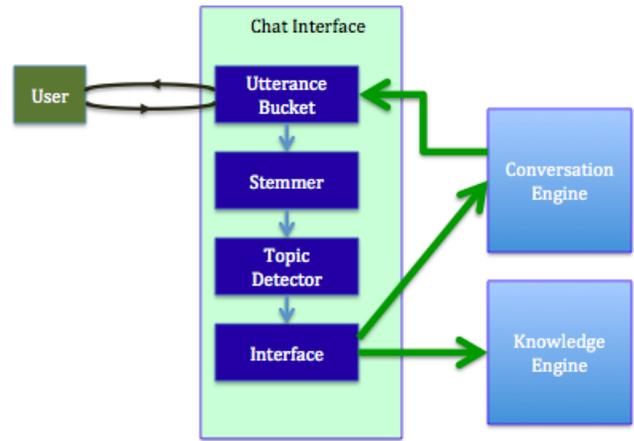


Fig. 4. Chat Interface: This module directly interfaces with the user. The blue block arrows show flow of information between the sub-modules within the Chat Interface. The green block arrows indicate flow of information between the modules external to it.

The Stemmer module reduces the text to the root stems. Stemming is a process by which inflected words are reduced to their root form. 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 [9]. Then the stemmed set of words is subjected to the Stanford NLTK parser, that is part of the open source NLTK suite. This extracts a set of discerning keywords.

#### B. Knowledge Engine

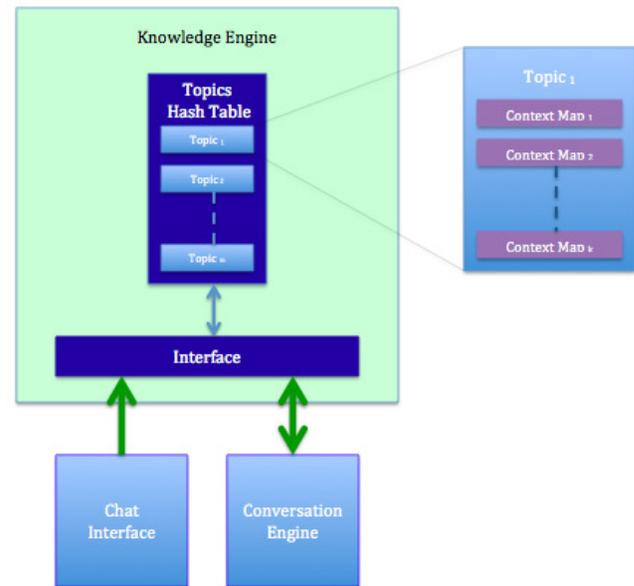


Fig. 5. Knowledge Engine: Contains the Speech Acts Hash Table, and the Topics Hash Table. Each Topic consists of Context Maps. The blue block arrows show flow of information between the sub-modules within the Chat Interface. The green block arrows indicate flow of information between the modules external to it.

The Knowledge Engine has a Topic Hash Table that organizes the set of topics belonging to the scope of the conversation of the chatter bot. Each topic is a broad area of material for the particular domain being implemented. Each specific topic entry in the hash table consists of a set of context maps that encode specific information about the topic. Currently, the context maps are realized using several goal fulfillment maps.

### C. Conversation Engine

The Conversation Engine module controls the direction of the conversation, depending on the criterion pre-coded in the goal-fulfillment maps selected by the Knowledge Engine. This directs the flow of the conversation.

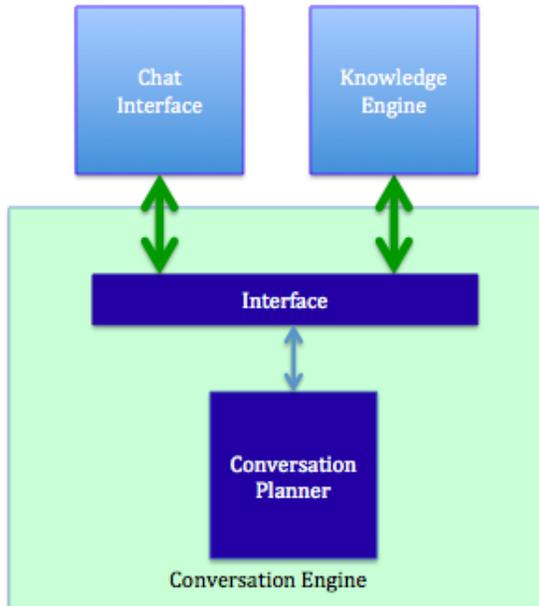


Fig. 6. Conversation Engine: Contains the Probabilistic Finite State Automaton, and the Conversation Planner. The blue block arrows show flow of information between the sub-modules within the Chat Interface. The green block arrows indicate flow of information between the modules external to it.

## IV. EVALUATION CRITERIA

It is important to have a standard, consistent metric to measure the quality of conversations. We borrow ideas from the theory of pragmatics to define some metrics to evaluate conversations. 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 [10]. It studies how the transmission of meaning depends not only on the linguistic knowledge (e.g., 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 [11].

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' 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 [12].

The cooperative principle describes how people interact with one another. As phrased by Paul 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. 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; these principles 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 [13].

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:

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

Saygin et al [14] 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 chatter bot can be graded for a specific criterion.

## V. RESULTS

We show some sample conversations that our chatter bots are able to perform. These are actual traces of conversations that our chatter bot could perform when fed with simulated customer inputs. These conversations are of a troubleshooting nature, where the chatter bot tries to troubleshoot a customer service issue.

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

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

*Chatter Bot*: Is the margin higher or lower?

*Customer*: Lower.

*ChatterBot*: Do you have pending buy orders?

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

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

Conversation Transcript 2: Conversation between our chatter bot and a customer trying to resolve a margin issue.

We observe that unlike in Conversation Transcript 1, our chatter bot is able to successfully resolve the issue. Let us consider another conversation with our chatter bot from a different domain.

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

*Student*: I am unable to register for CS 351.

*Chatter Bot*: Have you taken CS 151?

*Student*: Yes.

*ChatterBot*: Have you taken CS 251?

*Student*: Yes, but I withdrew from the class before the final exam.

*Chatter Bot*: CS 251 with a B or better is a pre-requisite for CS 351.

Conversation Transcript 3: Conversation between our chatter bot and a student trying to resolve a registration issue.

Each of these conversations grade well against Grice’s four maxims. They satisfy the *quality* maxim since the responses are factually correct. They satisfy the *quantity* maxim, since the information given is adequate and not superfluous. They satisfy the relation maxim since the responses are relevant to the context of the conversations. Finally they satisfy the *manner* maxim since the responses are unambiguous and do not seed any doubt.

Hence, we can conclude that our chatter bots perform relatively better at least in the subdomain of trouble shooting common issues.

## VI. FUTURE WORK

We will incorporate ideas from speech act theory to further enrich the conversation capabilities of the chatter bots. Speech act theory asserts that with each utterance in a conversation, an action is performed by the speaker. These actions (or speech acts) are organized into conversations according to predefined patterns [15]. This is subtly distinct from the indirect speech acts theory described by Searle [16], [17]. Winograd and Flores [18], [19] show that conversation for action is an example of a pattern of speech acts organized together to create a specific type of conversation. Speech act profiling looks at conversations as existing empirical data that are categorized during or after they actually take place. This approach partially answers Goldkuhls [15] question: Can a more inductive way of investigating conversations be performed with less use of pre-defined communicative patterns by using existing conversations as data, profiling them, and, finally, automatically classifying them according to patterns of interest, patterns that are either predefined or created post hoc?

Speech act profiling [20] can analyze and visualize conversations and their participants. It extends the method of Stolcke et. al [21] of dialog act modeling and combines it with Alston’s [22] idea of illocutionary act potential, which is realized with Subasic and Huettner’s [23] fuzzy typing. These methods are used to create a set of summed probabilities for each speech act type during a conversation, which are then subtracted from the training corpus average to obtain divergences from normal speech. The speech acts used are the 42 dialogue acts in the modified SWBD-DAMSL tag set [24]. With the speech act potential probabilities and the categories of speech acts defined, a representation of the speech act profile is created.

Conversations with each speech act manually annotated are manually categorized, and the conversations from each category used to train the corresponding hidden Markov model in a process similar to the one used in speech act profiling described in [20] and [21]. New conversations to be classified are run through each hidden Markov model using the forward-backward algorithm for hidden Markov Models [25]. The classification represented by the model yielding the greatest probability is assigned to the conversation.

## VII. CONCLUSION

Our language model is based on Grice’s maxims, and is implemented with our conversation architecture that enables chatter bots to hold meaningful conversations in the customer service domain. We have shown that our chatter bots can perform well in conversations where they are required to troubleshoot issues. Our chatter bots are able to go beyond mere question-answer or utterance-response exchanges and hold a more meaningful conversation with a human.

## ACKNOWLEDGMENT

We thank Dr. Chuck Wooters, International Computer Science Institute, Berkeley, CA for ideas and discussion.

## REFERENCES

- [1] Gartner, "Organizations that integrate communities into customer support can realize cost reductions of up to 50 percent," Tech. Rep., March 2012. [Online]. Available: <https://www.gartner.com/it/page.jsp?id=1929014>
- [2] D. Michie and C. Sammut, *InfochatTM Scripters Manual.*, Convagent Ltd, Manchester, 2001.
- [3] K. O'Shea, Z. Bandar, and K. Crockett, "A novel approach for constructing conversational agents using sentence similarity measures," in *World Congress on Engineering, International Conference on Data Mining and Knowledge Engineering*, 2008, pp. 321–326.
- [4] Y. Li, "A method for measuring sentence similarity and its application to conversational agents," in *Florida Artificial Research Society Conference*, 2004.
- [5] C. Sammut, "Managing context in a conversational agent," *Electronic Transactions on Artificial Intelligence*, vol. 3, no. 7, pp. 1–7, 2001.
- [6] K. O'Shea, Z. Bandar, and K. Crockett, "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, November 2009, pp. 92–99.
- [7] —, "Towards a new generation of conversational agents using sentence similarity," *Advances in Electrical Engineering and Computational Science, Lecture Notes in Electrical Engineering*, vol. 39, pp. 505–514, 2009.
- [8] K. O'Shea, Z. Bandar, and K. Crockett, "A conversational agent framework using semantic analysis," *International Journal of Intelligent Computing Research (IJICR)*, vol. 1, no. 1/2, March / June 2010.
- [9] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.
- [10] J. L. Mey, *Pragmatics: An Introduction.*, 2nd ed. Oxford: Blackwell, 2001.
- [11] P. Grice, "Meaning," *The Philosophical Review*, vol. 66, no. 3, July 1957.
- [12] —, "Logic and conversation," *Syntax and Semantics*, vol. 3, pp. 41–58, 1975.
- [13] —, *Studies in the Way of Words.* Harvard University Press, 1989.
- [14] A. P. Saygin and I. Ciceklib, "pragmatics in human-computer conversation," *Journal of Pragmatics*, vol. 34, pp. 227–258, 2002.
- [15] G. GoldKuhl, "Conversational analysis as a theoretical foundation for language action approaches?" in *Proceedings of the 8th international working conference on the languageaction perspective on communication modelling.*, H. Weigand, G. GoldKuhl, and A. de Moor, Eds., Tilburg, The Netherlands., 2003.
- [16] J. Searle, *Speech Acts.* Cambridge University Press, 1969.
- [17] —, *Indirect speech acts*, ser. Speech Acts. New York: Academic Press, ch. 3, pp. 59–82.
- [18] T. Winograd, "A language/action perspective on the design of cooperative work," *Human Computer Interaction*, vol. 3, no. 1, pp. 3–30, 2003.
- [19] T. Winograd and F. Flores, *Understanding computers and cognition: A new foundation for design.* Norwood, New Jersey: Ablex Publishing Corporation., 1986.
- [20] D. P. Twitchell, M. Adkins, J. F. Nunamaker, and J. 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.
- [21] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. V. Ess-Dykema, and M. Meteer, "Dialogue act modeling for automatic tagging and recognition of conversational speech," *Computational Linguistics*, vol. 26, no. 3, pp. 339–373, 2000.
- [22] W. P. Alston, *Illocutionary acts and sentence meaning.* Ithaca, N.Y.: Cornell University Press., 2000.
- [23] P. Subasic and A. Huettner, "Affect analysis of text using fuzzy semantic typing," in *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, 2001, pp. 483–496.
- [24] D. Jurafsky, L. Shriberg, and D. Biasca, "Switchboard swbd-damsl shallow-discourse-function annotation coders manual," University of Colorado - Boulder, Tech. Rep., February 1997.
- [25] L. R. Rabiner, "a tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, February 1989.