

Fundamental Challenges of Creating a Perfect Chatbot

Oliver Shi

Dr. Mark Liberman

12.16.22

Thesis submitted to meet CIS 4980 BAS graduation requirement

Abstract

As the modern world becomes increasingly digitalized, we find ourselves interacting with chatbots on a near daily basis. As these chatbots become increasingly pervasive, they have a growing influence on our lives. Chatbots such as Apple's Siri or Amazon's Alexa are passively listening to our conversations and use our data for various functions including displaying targeted advertisements. Newly released chatbot, ChatGPT, has redefined the realm of use cases for chatbots. Thus, with the growing influence of chatbots, it is important to understand the fundamental limitations of chatbots due to both human psychology and conversational theory.

Table of Contents

ABSTRACT	2
1 INTRODUCTION.....	4
2 EVALUATING CHATBOT SUCCESS.....	5
2.1 TURING TEST	5
2.2 MODERN TESTS.....	5
2.3 TESTING LANGUAGE MODELS USING LANGUAGE MODELS?	6
3 HUMAN LANGUAGE CHALLENGE	7
3.1 SEMANTICS	7
3.1.1 <i>Ambiguity</i>	7
3.1.2 <i>Analogies</i>	9
3.1.3 <i>Binary Relations</i>	10
3.2 IMPLICATURE	11
3.2.1 <i>Introduction to Implicatures</i>	11
3.2.2 <i>Interpreting Binary Implicatures</i>	11
3.2.3 <i>Conventional and Conversational Implicatures</i>	13
3.2.4 <i>Grice's Cooperative Principle</i>	13
4 MACHINE MODEL CHALLENGE	17
4.1 DIALOGUE BREAKDOWN.....	17
4.1.1 <i>Causes of Dialogue Breakdown</i>	17
4.1.2 <i>Dialogue Breakdown Challenge</i>	17
4.1.3 <i>Repairing Dialogue Breakdowns</i>	18
4.2 THE ALIGNMENT PROBLEM.....	18
5 COMPLICATED INTERACTIONS	20
5.1 CURRENT CHATBOTS	20
5.2 EXTERNAL REPRESENTATION OF STATE.....	22
5.2.1 <i>The Dialogue State Tracking Challenge</i>	23
5.2.2 <i>Statistical Models</i>	27
5.2.3 <i>Discriminative Models</i>	28
5.2.4 <i>Overall Issues</i>	30
5.3 STYLE, TONE, AND BIAS	31
6 ETHICAL CHALLENGES	32
7 FUTURE OUTLOOK AND CONCLUSION	34
REFERENCES	35

1 Introduction

A conversational agent or chatbot is a computer program that simulates conversations with users. They are utilized to answer questions and provide information to users that mimic a human. Chatbots need to understand natural language, process requests or utterances, and respond with a relevant utterance. Inputs can be both text and voice in different languages.

Modern chatbots are extremely powerful. Some of the most well-known chatbots include Siri, Alexa, and Google Assistant. They are capable of handling simple question-answer requests such as setting reminders, ordering products, or sending emails.

ChatGPT is a newly released chatbot powered by OpenAI that was made available to the public on November 30th, 2022. It utilizes a large language model (LLM) based on GPT-3.5 architecture [21]. ChatGPT interacts with humans in a conversational way with a format that allows for sustained conversation as well as a human-like tone throughout these conversations. ChatGPT was trained using reinforcement learning from human feedback with supervised fine-tuning.

This paper is a literature review on the limitations of modern chatbot technology. Section 2 will provide an overview of the methods of evaluating chatbot success. Section 3 will examine the challenge of understanding and mimicking human language. Section 4 will cover challenges related to modeling human interaction and aligning chatbot goals with human intentions. Section 5 will delve into the challenges of holding complicated interactions that require knowledge of conversation history. Finally, section 6 will explore the ethical challenges that chatbots face.

2 Evaluating Chatbot Success

Before we can analyze the challenges of creating a more optimal chatbot, we must describe how we can evaluate the quality of a chatbot. Unlike algorithms, we can't simply evaluate a chatbot on efficiency or runtime. There is also no great way to evaluate accuracy since there are numerous "correct" answers to different requests. Responses are not binary, right or wrong, like other machine learning techniques such as image detection. While there are metrics we can use such as activity volume, bounce rate, retention rate, conversation length, goal completion rate, and other key performance indicators, they are not a holistic way to measure chatbot success [24]. Thus, other standards have been created to evaluate chatbot performance.

2.1 Turing Test

A classic test used on intelligent programs is the Turing Test. It tests if a computer is capable of showing behavior similar to humans. Specifically, it creates an imitation game where there is a machine and a human. There is an interrogator who blindly asks questions to both the machine and the human. For a machine to pass the Turing Test, the machine must convince the interrogator that it is the human. Concurrently, the human is trying to help the interrogator correctly identify the machine [5].

Critiques of the Turing Test argue that machines can pass it given a very specific context. Additionally, people think that the Turing Test is not sufficiently demanding. An early chatbot – ELIZA – passed a "restricted Turing test for machine intelligence" [16]. However, we know that ELIZA quickly fails when asked complex questions. Thus, the Turing Test is a seemingly outdated method for evaluating chatbot performance.

2.2 Modern Tests

Since the creation of the Turing Test, chatbots have significantly improved. In niche contexts, chatbots can emulate near human performance. Thus, there needed to be new methods to evaluate chatbot performance. Before 2018, there was no multi-turn systematic methodology for evaluating chatbots. In 2018, ChatEval was created as a scientific framework for evaluating chatbots. ChatEval utilizes both automatic and human components to grade chatbot performance. Automatic evaluation metrics use "lexical diversity, average cosine-similarity between word embeddings and responses, BLEU-2 score, and response perplexity" (Sedoc, 2019) [24]. Lexical diversity is the number of n-grams divided by generated tokens, representing the uniqueness of responses. Average co-sine similarity compares the differences between generated responses and ground truth responses (the expected response). BLEU-2 or bilingual evaluation understudy compares translations from the target goal and the actual response [31]. Response perplexity measured the likelihood that models would predict the correct response.

ACUTE-EVAL was introduced in 2019 and it compares different dialogue models. However, instead of having a human connect with a chatbot, ACUTE-EVAL utilizes chatbot to chatbot conversations and self-chats, where a chatbot model converses with itself. Models were

then graded on “engaginess, interestingness, knowledge, and humanness” [15]. These results would be compared across models and compiled into different tables.

2.3 Testing Language Models Using Language Models?

When testing the success of a chatbot, we use different language models to weigh different outcomes and assign them scores. To assign them scores, we create an optimal language model based on the input data provided. This optimal model is then compared to the language model that the chatbots utilize. Thus, it is implied that a perfect chatbot will output results that are identical to the language model used to codify the evaluation. However, we know that our language models themselves are imperfect, and thus having them as the optimal solution will result in artifacts.

In practice, we model human languages using statistical distributions. The rank-frequency law states that “the frequency of a word in a corpus decays exponentially in the frequency rank of that word... so that it follows the power-law distribution” [17]. The type-token law states that as a document becomes longer, the number of new unique tokens will decrease as a function of a Poisson distribution. Given these two laws, tendencies of length, and stopword frequency, we can compare two different language models using statistical interference tests that are based on these metrics.

Taking random samples S1 and S2 from each language model, we can test to see if their calculated distance is not equal to zero with statistical significance. Using these tests, we can estimate the similarity between two different natural language models given samples (in this case outputs from a chatbot that highlights the chatbot model and an optimized response distribution, which is the basis language model). An issue is that results are highly dependent on the model’s architecture and what sample is selected. Because a sample is a small subset of the overall natural language, data can easily be skewed in one direction or another.

These are just a few of the many modern ways to evaluate chatbots. However, there is still no perfect way to evaluate a chatbot, and thus no way to create the perfect chatbot. There exist two critical flaws for most models. The first is that models are reference-based, meaning they evaluate chatbots relative to an optimal/expected conversation. In reality, conversations are incredibly open-ended and there is a multitude of paths and branches that conversations can traverse. Thus, it is difficult for the references to sufficiently cover all possible paths.

3 Human Language Challenge

3.1 Semantics

3.1.1 Ambiguity

The first fundamental challenge of optimizing a chatbot lies in decoding the human language. Semantic analysis in natural language processing examines the relationship between different words within a sentence to comprehend the overall meaning. In human conversation, ambiguous words are extremely common. Semantics allow machines to contextualize these words to create meaningful recognition.

For example, take the sentence

I saw the Golden gate bridge flying into San Francisco [4]

For humans, it is clear that the subject was flying into San Francisco and saw the Golden gate bridge while he/she was flying. However, without context, the sentence could be interpreted as the subject saw the bridge itself flying.

Other common examples include

I ate dinner with a friend

I ate dinner with a fork [4]

Humans depend on the syntax of the language to decide how words are combined to make a larger meaning. Machines struggle to decipher context without grammar trees and syntactical models. Algorithms are constrained to process only the information they can see in text or voice data and thus cannot consider the background information that humans do. Natural language processing aims to respond to some language input. Machines rely on a formal language that is unambiguous, has simple rules of interpretation and inference, and has a consistent logical structure determined by the sentence's form. Thus, a challenge for chatbots is to convert natural language into formal machine language [4].

This challenge is difficult because in natural language, words have ambiguity, and the same word will have different meanings in varying contexts. While we can create models for language using constraints, these are not always accurate. A popular example of structural parsing is creating a grammar tree and using context-free grammar to parse natural language. Sentences are broken down into NP and VPs before being further deconstructed. Another approach is using a top-down parsing approach where sentences are searched from the root node with constant substitutions at each level. See figure 1 for an example of top-down parsing for the utterance “book that flight” [4].

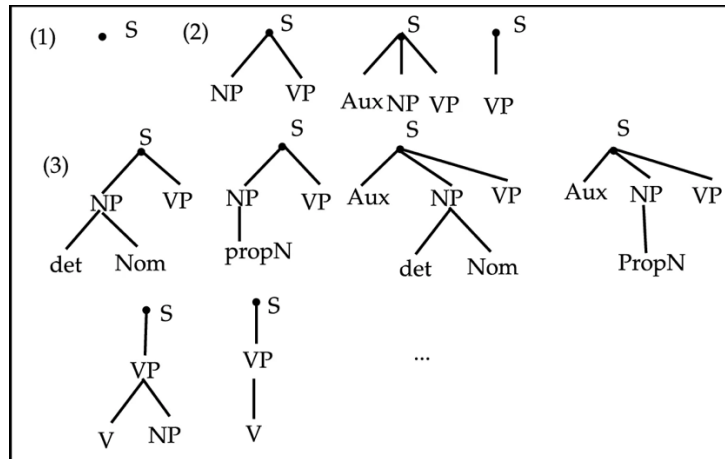


Figure 1: Top-Down Parsing Visualization

We see that at the first level there is a root node, *S*. This node is broken into 3 sub-trees at the second level. At the third level, these subtrees are expanded and compared to the utterance, “book that flight”. We see that the trees of form *S* -> *NP VP* and *S* -> *Aux NP VP* will not match our utterance. Thus, we try *S* -> *VP*, and find that we can match book that flight with form *V*, *NP*. While this example was simple, we see that algorithms needed the ability to break down each word and match it with a part. However, this becomes much more difficult when models encounter ambiguity. How does the model understand which words can be classified as what form, and thus how can the model process the overall utterance?

One way to address the challenge of ambiguity is probabilistic parsing. Probabilistic parsing is like top-down parsing with non-terminal nodes that have values that recognize the syntactic and semantic relations. Restrictions are relaxed so that every possible analysis of a sentence has a nonzero acceptability. This means that this method can exhaustively try all alternatives. Thus, we create a probabilistic context-free grammar where each parse-tree has some probability of being correct. While not complete, this provides a probabilistic language model which better addresses ambiguity.

To highlight a simple example of ambiguity in chatbots, I asked popular chatbots *Siri*, *Google Assistant*, and *Alexa* the following question:

Send an email with Personal: I want to send an email from my personal email account

Siri

Siri incorrectly composed an email with my personal email account in the *To* field

Google Assistant

Google incorrectly directed me to results from a Google search

Alexa

Alexa was not able to compose an email.

In this example, the object is “Personal”. However, it is ambiguous whether I want to my personal email address as the sender or recipient. Using common sense, I would want to compose an email with my personal email address as the sender. However, the request confused Google Assistant and Alexa while Siri misinterpreted my request and drafted an email with my school email as the sender and my personal email as the recipient.

This ambiguity results in a clear challenge for chatbots to overcome. A chatbot should ask for clarification when faced with an ambiguous request. We see this when asking Siri

Set a reminder for tomorrow at 9am

If asked late at night after 12am. Siri will ask for clarification, stating:

Did you mean: today, December 12th at 9am or tomorrow, December 13th at 9am?

However, this is not a consistent response and at other times can cause user frustration. IBM Watson utilizes disambiguation, which is the process of clarifying the meaning of sentences with follow up questions. This occurs when there are two or more nodes that trigger disambiguation, requiring the confidence scores of multiple intents to be close in value to the top intent and for the top intent to have a confidence score above 0.2 [34]. However, this method causes consumer confusion at times and does not consistently produce accurate responses.

For ChatGPT, even small tweaks in phrasing can result in the model claiming to not know the answer when a slight rephrase would lead to a correct response [21]. While chatbot technology has improved in addressing the issue of ambiguity, there is still significant progress to be made. The subtleties of human language and phrases confuse even the most modern models.

3.1.2 Analogies

If I ask a chatbot to send a message to convey that I love my mom like a rose, I am comparing my emotions for my mother to the beauty and elegance of a rose. This analogy relies on figurative language as I do not literally love my mom like a rose, but instead, I utilize the rose as a metaphor to describe the beauty and pureness of my emotions.

NLP has difficulties processing analogies. A formal analogy is when a relationship from a string of words $x : y$ is applied to z to get a new string of words that follows that same relationship. An example is $abc : aabbccd :: efg : eeffggh$. In plain English this could be $white : black :: up : down$.

Formal analogies can be solved using vector arithmetic. Given $A : B :: C : x$ where we want to solve for x , we can use vector v

$$v = \vec{B} - \vec{A} + \vec{C}$$

We then define x as

$$\vec{x} = \arg \max_w \cos(\vec{w}, v),$$

Afterward, we can solve for x given cosine similarity between words. This is useful to help chatbots interpret analogies when spoken by humans [14].

2.1.3 Binary Relations

To address the semantics issue, NLP knowledge bases have been created that aggregate the different kinds of relations between words. Examples of these semantic relations include REFER, INVOLVED_AGENT, INVOLVED_LOCATION, REASON, etc. By leveraging this knowledge base, NLP algorithms can more accurately predict the relationships between different words. These bases for NLP ontologies rely on relationships to convey lexical meaning and interpretations [22].

In ontologies, these relationships are typically asymmetric. For example, a cheeseburger's relationship with cheese and bun may be that a cheeseburger is comprised of cheese and buns. However, the other direction is not correct: cheese and buns are not comprised of cheeseburgers. This is challenging because the path from a target to a source is different from the source to target. This means that in analysis, databases need to store the entire unidirectional map. This leads to an exponentially growing storage space requirement with respect to sentence complexity and length.

While storage space is not a large concern, it is relevant for designers of chatbots when considering costs.

The current limits of our natural language models make it difficult for chatbots to accurately process natural language. Language is often ambiguous with the same word having different meanings in different contexts. It is hard to understand the intentions and motivations behind sentences, making deciphering meaning even more difficult. The subtleties of human language in various spoken and written languages make accurate parsing very challenging.

3.2 Implicature

3.2.1 Introduction to Implicatures

A second issue with the human language is implicatures. An implicature is “the act of meaning or implying one thing by saying something else of the object of that act” [6]. There are four main kinds of implicatures: conventional implicature, conversational implicature, generalized conversational implicature, and particularized conversational implicature. An example of an implicature is

I ate some of the cookies.

There are four ways implicatures can be derived from this statement. The person could have eaten all of the cookies, not have eaten all of the cookies, does not know who ate the cookies, or knows who ate all of the cookies. The speaker conveys meaning beyond what is just said and relies on background knowledge. While humans can interpret what the speaker truly means with context, this is much more challenging for machines.

H.P. Grice coined the term implicature when a speaker says p but implicates q . Interestingly, the same statement p can implicate different q because of different implicatures which may depend on the preceding information. For the listener to understand a speaker’s implicatures, we assume that the speaker follows Grice’s four maxims. These maxims, known as the Gricean Maxims are:

- The maxim of quantity: people will share as much information as necessary.
- The maxim of quality: people will be truthful.
- The maxim of relation: people will share relevant information.
- The maxim of manner: people will be clear and avoid ambiguity [23].

However, speakers often do not fully adhere to these maxims and will produce inadequate information. Thus, speakers and listeners need to have cooperative conversations to add clarity to any potential points of confusion. This means that the listener must adapt to what the speaker is sharing. In the context of natural language processing, this creates a significant challenge for machines to infer implicatures with incomplete information and induces a high processing cost.

3.2.2 Interpreting Binary Implicatures

Modern chatbots understand implicatures to some extent. For example, if I ask, “should I wear a jacket tomorrow”, Siri will respond with the low temperature of the day, showing that it understands I am implying that I want to know the weather. However, in other cases, chatbots are not as adept in processing implicatures. Stating “I’m going to forget to buy eggs tomorrow” results in a reminder to buy eggs set for the next day. However, the similar statement of “I’m worried about forgetting to buy eggs tomorrow” prompts Siri to state that they are sorry I am going through this and share positive affirmation. Google simply brings up a Google search with both statements while Alexa can set a reminder for both.

So how do machines interpret implicatures? There have been various attempts to codify implicatures in natural language processing. One strategy is inferring implicit opinions that are either positively or negatively affecting other entities. This utilizes a goodFor/badFor (gfbf) framework with events. We can use Integer Linear Programming with gfbf events as variables and interdependencies that come from implicatures as constraints [8].

Given a sentiment towards a , which is b , b is either positive or negative. Thus, with some combination of gfbf events as a , we can create a table (see Table 1) of what combination of sentiments and events implies what true meaning [8].

	s(gfbf)	gfbf	→	s(agent)	s(theme)		s(gfbf)	gfbf	→	s(agent)	s(theme)
1	positive	gf	→	positive	positive	3	positive	bf	→	positive	negative
2	negative	gf	→	negative	negative	4	negative	bf	→	negative	positive

Table 1 (Deng and Wiebe, 2014).

Now, the language model needs to assign labels to each variable in order to understand the implied sentiment. We can use the following method to do so. For each variable, k , in the *GFBF* schema, it has agent and theme variables, k, j which come from the Entity. These are combined to form triple units $\langle i, k, j \rangle$ with the goal of giving the optimal labels to the entity and gfbf variables. We can thus now assign the ILP with an objective function of

$$\min_{u_{1gf}, u_{1bf} \dots} \left(-1 * \sum_{i \in GFBF \cup Entity} \sum_{c \in L_i} p_{ic} u_{ic} \right) + \sum_{\langle i, k, j \rangle \in Triple} \xi_{ikj} + \sum_{\langle i, k, j \rangle \in Triple} \delta_{ikj}$$

which is subject to

$$u_{ic} \in \{0, 1\}, \forall i, c \quad \xi_{ikj}, \delta_{ikj} \in \{0, 1\}, \forall \langle i, k, j \rangle \in Triple$$

These variables have constraints which represent machine interpretations of implicatures.

$$\sum_{c \in L_{GFBF'}} u_{kc} = 1, \forall k \in GFBF \quad (3)$$

$$\sum_{\substack{i \in Entity \\ \langle i, k, j \rangle \in Triple}} \sum_{c \in L_{Entity}} u_{ic} = 1, \forall k \in GFBF \quad (4)$$

$$\sum_{\substack{j \in Entity, \\ \langle i, k, j \rangle \in Triple}} \sum_{c \in L_{Entity}} u_{jc} = 1, \forall k \in GFBF \quad (5)$$

As a result of this, we can give each variable a label of *gf* or *bf* and possibly reversed for negated events. The entities will also be labeled positive or negative given the sentiment toward the events. Thus, this method can decipher implicatures in a binary positive or negative way with the addition of sentiments towards objects in such utterances.

While we know that integer linear programming is NP-Hard, we can arrive at a “good enough” solution when running this. Solving for pairs $\langle i, j, k \rangle$, we can arrive at a conclusion of

whether the relationships are goodFor or badFor. Applying this to chatbots, our conversational agents can attempt to understand implicatures, where speakers are negating information that they are sharing. Additionally, while not the topic of this paper, it helps with sentiment analysis of users.

A practical example would be sharing with a chatbot “The revised plan should curb my transportation costs over the next year”. The goal would be for the chatbot to respond with an analysis of transportation costs and potentially suggest options as to how to cut the costs in the next year. However, we see that the word costs tend to imply negative sentiment. Plan is a positive agent. Thus, by combining these, the chatbot would find a positive, badFor event where our plan is badFor transportation costs. The overall sentiment is positive because we are happy with lower costs. So, the chatbot with a goal to satisfy the user will find positive plans that are bad for the costs (making it lower).

3.2.3 Conventional and Conversational Implicatures

We can see that with some modern techniques, chatbots can interpret simple implicatures in good for and bad for. However, implicatures go far beyond that as previously discussed. A conventional implicature is one that is generally understood without any additional context. For example, stating “even Ken knows it’s unethical” [13] implies that Ken is the least likely to know something is unethical. The word *even* and similar ones like *but*, *therefore*, *still*, *again*, *etc.* are known to suggest additional meanings in a sentence that are not explicitly stated without the need for additional context. To understand conventional implicatures, they need to have a deep understanding of a language and the context in which specific words are used.

Conversational implicatures are implicatures where the meaning is implied from the context of the current conversation. Figurative and expressive language is a part of conversational implicatures that are dependent on the context of words in the real world. These include hyperboles, irony, overstatement, understatement, and metaphor. For example, one could ask a chatbot “I’m looking for some good eats” which implies that they want recommendations for a restaurant this evening. To identify conversational implicatures, chatbots analyze commonly used phrases and their intended meanings. Conversational implicatures can be difficult to understand because they often are implicit and require an understanding of the broader conversation or the world around them. Additionally, they can be very subjective and differ for each user.

3.2.4 Grice’s Cooperative Principle

Not only do chatbots have issues interpreting implicatures, but they also need to ensure that they follow the Gricean Maxims when responding to users. This is to ensure that the user fully understands what the system is sharing, and thus we must ensure that the systems provide adequate context. Grice’s cooperative principle is that participants in the conversation will have “conversational contribution such as is required, at the stage at which it occurs, by the accepted

purpose or direction of the talk exchange” (Grice 1975) [10]. In other words, they will try their best to follow the four Gricean maxims.

This leads to two challenges that chatbots face when dealing with natural language. Firstly, chatbots must sometimes interpret statements made by others that do not follow the cooperative principle, which leads to inaccuracy. Secondly, chatbots must respond to statements with outputs that do follow the cooperative principle.

Here are some examples of violations of the four Gricean maxims.

Quantity: there will be as much information shared as necessary

User: “Where is my CIS 4980 class tomorrow?”

Chatbot: “On campus”

While true, the chatbot violates the quantity maxim because it is obvious that class is on campus. The user is asking for a specific room number and building while the chatbot responds with a general location. Thus, it is not providing sufficient information to help the user.

Quality: the truth will be shared

User: “Where is my CIS 4980 class tomorrow?”

Chatbot: “You do not have CIS 4980 tomorrow”

The chatbot violates the quality maxim because it is sharing a lie. The user has CIS 4980 tomorrow and is expecting to receive a truthful response.

Relation: responses will be directly related to the question

User: “Where is my CIS 4980 class tomorrow?”

Chatbot: “You have a workout scheduled at 3pm tomorrow”

The chatbot violates the relation maxim because the user is asking for the location of his computer science class tomorrow and the chatbot responds with information that is irrelevant. The user is not inquiring about his workout class nor the time of it.

Manner: responses will be clear, concise, and unambiguous

User: “Where is my CIS 4980 class tomorrow?”

Chatbot: “Your Computer Information Science class four thousand nine hundred and eighty on December 12th, 2022 is located a few blocks to your East in the University of Pennsylvania’s...”

The chatbot violates the manner maxim because it provides an overly verbose response that is not necessarily clear.

Interestingly, individuals will sometimes flout a maxim – purposefully violating one or more in an obvious way that is clear to the listener. This can result in an implicature because it is expected that the listener will understand the hidden meaning behind words. For example,

User: “what Chinese restaurants are currently open in Philadelphia?”

Chatbot: “Philadelphia, Pennsylvania?”

User: “no, Philadelphia, Uzbekistan”

The user is being sarcastic and is implying that of course, he is looking for restaurants in Philadelphia, Pennsylvania. This illustrates the importance for chatbots to understand both how to follow Gricean Maxims as well as what a purposeful violation of the maxims signifies [32].

A chatbot’s similarity to a human conversational agent is related to their adherence to the cooperative principle. In a study where humans conversed with AI chatbots, their unexpected delays in response were measured. These delays were attributed to additional processing due to confusion from chatbot utterances. The chatbots would purposefully violate one or more of Grice’s maxims. In the table below, we can see the results of response delay in relationship with violations of the quantity, quality, and manner maxims [2].

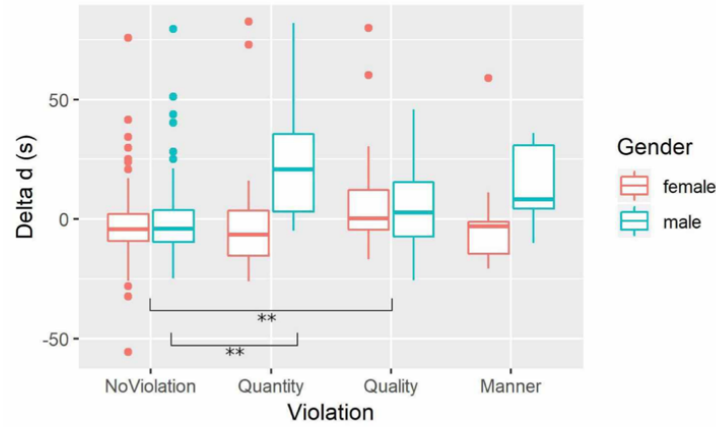


Fig. 1. Δd in first conversation. Box plot of Δd in seconds when participants spoke to the *artificial agent* in the first conversation.

** $p_{Holm} < 0.01$

Figure 2: Box plot of Δd , the difference between observed and expected delay, in seconds when participants spoke to the artificial agent [2].

From figure 2, we see that when chatbots violate a Gricean maxim, there is a noticeable response delay from the participant. When asked to identify the chatbot versus a human, a violation of a Gricean maxim led to the correct identification of the chatbot that was significantly higher than when a Gricean maxim was not violated [2]. Thus, we see that it is important to create chatbots that can respond with utterances that abide by the cooperative principle.

Overall, processing implicatures is a two-pronged complex task that requires a deep understanding of language and context. It involves analyzing the words used and the context in which they are used and inferring the intended meaning based on this analysis. Even for humans, this can be a challenging task. Because chatbots are limited by the data they have been trained on and lack contextual information, they have difficulty accurately processing implicatures. On top of understanding implicatures, chatbots must produce responses that adhere to the cooperative principle so that the implicatures it shares can be understood. The maxims of quantity, quality, relation, and manner serve as a guide for efficient and effective communication while violation

of these maxims causes confusion and misinterpretation. Chatbots are machine learning models and do not have the inherent ability to communicate in the same way that humans do. They cannot follow the constantly changing social rules or principles of communication like a human would.

4 Machine Model Challenge

4.1 Dialogue Breakdown

4.1.1 Causes of Dialogue Breakdown

One of the most concrete ways of thinking of a chatbot is dialogue breakdown. A dialogue breakdown is a situation where two or more individuals in a conversation are unable to continue the conversation as a result of the dialogue. Chatbots need to respond to prompts with utterances such that the conversation has the ability to continue. It should be noted that humans often have dialogue breakdowns in real life, but chatbots tend to have them far more often.

In discussing dialogue breakdowns, we will first examine common errors made by chatbots that cause them. At the highest level, there are utterance-level errors. They include but are not limited to:

- 1) Violation of Form: the language itself is violated, for example responding to an English conversation in a different language.
- 2) Uninterpretable: the utterance is unable to be interpreted as a word, for example responding with “tmeai”
- 3) Grammatical error: the grammar of the language is violated to the extent that it is missing one or more elements that make it a sentence, for example missing a subject or verb.

The next level of dialogue breakdown is a violation of content. These contain correct words that form grammatically correct sentences but are incorrect in different ways. These include:

- 4) Semantic error: where arguments do not make sense, for example, “my computer is hungry.”
- 5) Wrong information: where responses have clearly incorrect information such as “I walked over from China to Mars”

The following utterance-level issues are response-level errors. These include violations of form where utterances do not continue the functions of the previous conversation. These include ignoring questions, ignoring requests, ignoring proposals, and ignoring greetings. There is also ignoring expectations, where there is an expected follow-up response that is missing.

A further level deeper lies context-level errors. These are when chatbots respond with utterances that are out of context or lack continuation between topics. These include unclear intentions, topic transition errors, a lack of information, self-contradiction, contradiction, and over-repetition.

The deepest level of errors lies with society level errors. When utterances do not properly consider controversial subjects, such as hate speech or violence, they go against what is universally agreed upon [12].

4.1.2 Dialogue Breakdown Challenge

Given these errors that cause dialogue breakdowns, the next challenge is learning how to detect dialogue breakdowns to prevent chatbots from sharing utterances that result in a dialogue breakdown. Some researchers have put forth the *Dialogue Breakdown Challenge* where systems share utterances and challengers must create an algorithm to detect when there is a dialogue

breakdown. Initial trials found that the best accuracy was around .643, which is not much better than randomly guessing (which had an average of .481). The highest precision was .531 and the highest recall was .783, compared to a baseline of .355 and .136, respectively. Thus, it was found that it is exceedingly difficult to detect dialogue breakdowns with accuracy [12].

A follow-up to this challenge was conducted 5 years later called the Dialogue Breakdown Challenge 4. From this challenge, it was assumed that, in English, detecting dialogue breakdown with machines has an upper bound of accuracy of around .7809 [12].

4.1.3 Repairing Dialogue Breakdowns

Thus, instead of trying to fully prevent dialogue breakdowns, a suggestion for chatbots is to repair dialogue breakdowns when they occur. When users respond to chatbots with confusion or even do not respond, there is a likelihood that a dialogue breakdown has occurred. There are different ways that chatbots can approach fixing this.

The first, and easiest strategy is to simply ignore the breakdown. The chatbot should continue the conversation as if nothing happened, and if anything, the user should fix the breakdown themselves.

However, if there is evidence of a breakdown, the chatbot can indicate that a breakdown is occurring and ask for clarity. An example would be asking “if ...” is what the users want. Similarly, a chatbot could state that it did not understand what was just shared and would repeat key phrases that would hopefully result in the user responding in a manner so that the chatbot can better process what has happened and respond properly. One of the most common strategies though is to simply transfer the chat to a human agent and allow it to continue there [1].

4.2 The Alignment Problem

Machine learning theories are fundamentally different from how humans interact and think. A machine cannot create a perfect model of human interaction. They struggle to form accurate theories of what people want and do not want. Thus, beyond NLP and conversations between chatbot and user, there is the underlying issue that machine models tend to not correctly model what humans genuinely want.

For example, if I want to book a plane ticket home for Christmas break, I would hope to find a ticket where I leave from Philadelphia to Dallas on some date between December 10th - 14th. I would also like to book a return flight from Dallas to Philadelphia after winter break on some date between January 7th – 10th. There are parameters that I prefer such as price, travel time, direct flight, comfort, etc. However, for the chatbot who may be assisting me throughout this process, this becomes a lot for it to process. A chatbot’s AI may simply be rewarding itself for booking me a flight, but not necessarily in a way that I or any other user would want, but in an efficient way that best satisfies its codified interests.

This is where a general chatbot’s lack of domain knowledge hinders it. A chatbot will not inherently have the knowledge of, in this example, what customers tend to prefer and how to balance these parameters of date, price, duration, etc. To people, this is known as qualitative

knowledge, such as learning that some individuals prefer price, while others prefer flying out on exact dates. There is “no universal way to bridge machine learning and qualitative knowledge, qualitative knowledge adds unique insights into the machine learning framework by using customized strategies” (Deng) [7]. Domain knowledge can be highly abstract or empirical. To train chatbots on such knowledge, we need to create large, labeled datasets to encourage active learning. These datasets are extremely intensive to create and verify, making overcoming this challenge difficult.

The alignment problem occurs when the goals and values do not align with the goals and values of the user, leading to poor communication between the chatbot and the user. This problem is complicated to address because chatbots operate in situationally dependent and complicated contexts with different users and expectations.

The first challenge is encoding values in our models. Developers face the technical challenge of how to formally codify our principles in chatbots such that they reliably act as they should. An extreme example of machine misalignment is when chatbots end up promoting abusive content such as Microsoft’s Tay [42]. This happened as a result of reward-hacking to reach its objective. Agents can discover disingenuous ways to achieve their codified goals.

The second challenge lies with what values we should encode in chatbots. While we would want chatbots to operate as optimally as possible, we must be wary of ethical concerns. It is important to keep in mind that machines do not inherently have ethics nor values. Thus, we must carefully align chatbots with our goals and encourage them to achieve them without breaking any ethical boundaries. For example, ChatGPT is built to refuse inappropriate requests, but will sometimes respond to harmful instructions [21].

5 Complicated Interactions

5.1 Current Chatbots

Machine models have a difficult time tracking the state of a user over a period of time. This is despite most chatbot users wanting to have longer conversations with chatbots as opposed to simple question and answer. For example, in the process of planning a trip, there are various factors that I am considering and would like to carry throughout a conversation. These include but are not limited to what day I leave, what day I get back home, my budget, what destinations I would like to see, etc. Human travel agents are capable of synthesizing this information and suggesting options that best fit their client's needs. For instance, a travel agent may suggest leaving a day early since this allows the client to save money, which is a high priority for them.

However, with chatbots, this process is very unproductive. Most publicly available chatbots are unable to start with understanding why will ask. When asking Expedia's chatbot to help me plan a trip, it simply directs the user to a human agent. Amtrak's chatbot, Julie, connects the user to the trip planning page but it is incapable of helping the user through the actual process. Similarly, Siri, Google, and Alexa all bring the user to a web search as opposed to directing them through the process of booking a trip.

WestJet has recently released Juliet, a digital assistant to help with travel bookings. As a powerful chatbot, Juliet asks the user what cities they want to travel between as well as when they want to travel. After processing this, Juliet takes the user to the WestJet website to book any potential trips. While this is a significant improvement, it lacks the expertise that human agents provide. Juliet and other chatbots would be unable to suggest, for example, leaving on a different date for a cheaper fare if that is what the user values because the conversational agents are unable to track data / state beyond the context of the conversation.

After a conversation with Juliet, it starts from the beginning and must relearn any information from the user. Additionally, it is unable to process a user's preferences and true desires of what they genuinely want. While it can maintain context between individual question and answer utterances, Juliet is only capable of asking a few pre-determined questions as opposed to responding to the user's inquiries.

Until recently, the best chatbots were designed for question and answer exchanges. However, they are incapable of holding a conversation where a specific context or state is kept throughout a conversation. As described above, customer service chatbots tend to be able to handle FAQ-like questions and responses. However, they are not able to process a conversation a level above that.

Chatbots like Alexa likely use large language models for its training yet are not capable of holding a conversation like ChatGPT can. While this might be due to a lack of technical ability, it is likely that there are other factors at play. LLMs are inherently unreliable, so Alexa would make frequent errors, causing people to stop using it. Similarly, nobody knows how to prevent LLMs to 100% refrain from insulting users or sharing false. Thus, by preventing Alexa from holding a conversation, Amazon reduces the risk of lawsuits that could result from Alexa's hurtful words. Moreover, Alexa's purpose is to interact with the real world like turning on lights,

ordering products, and more. If Alexa was capable of a full conversation, the expectations of users would be higher than achievable with present technology [38].

With the release of ChatGPT, users can have generalized conversations that go beyond simple question and answer. While ChatGPT retains chat history in a particular conversation, it does not have the ability to engage in complex conversations beyond providing answers to specific questions. While it can assist users by providing relevant answers to most of their questions, there is a lack of ability to engage in broader conversations beyond that. Moreover, unlike assistants like Siri, Alexa, and Google Assistant, ChatGPT is not able to interact with users beyond the conversation. ChatGPT cannot turn on lights, set reminders, compose emails, etc.

In human conversations, participants have background knowledge, common sense, and context to have an effective conversation. However, chatbots lack this capability and instead have typically “implemented with hierarchical rules in the forms of scripts” (O’Shea) [20]. Moreover, chatbots find difficulty adhering to Grice’s, especially those of quality and relation, which have to do with the context within a conversation.

ChatGPT has a broader range of context than previous chatbots, but its corpus is limited to the training data, which was from early 2022. It cannot access the internet, meaning any background relevant knowledge that came out after early 2022 is unattainable. It was trained on a large dataset of text, meaning it has useful knowledge on a wide range of topics. However, it lacks depth in topics, providing an overview in most cases instead of a deeper dive. Additionally, it lacks information specifically relevant to a user.

To have a successful conversation, chatbots need to have domain-specific information and a particular speech act. They provide the relevant information in the conversation such as user information and conversational context, respectively. Chakrabati and Luger have identified six acts for which conversation should be generated:

- 1) The Directive Speech Act: An utterance that acts as an instruction or command.
- 2) The Assertive Speech Act: An utterance that asserts a fact regarding the state of the world in the context of a conversation that involves all participants.
- 3) The Commissive Speech Act: An utterance that implies there will be a future change of state in the world in the context of a conversation.
- 4) The Expressive Speech Act: An utterance that expresses emotion or feelings about a conversation.
- 5) The Declarative Speech Act: An utterance that results in the state of the world to change that does not involve all participants.
- 6) The Goal-Fulfillment Speech Act: An utterance that results in the state of the world to be concluded as all issues in the conversation have been acknowledged [3].

Without the successful implementation of these acts, chatbots will not be able to hold conversations but instead will be constrained to QA. They cannot hold a conversation or understand if a conversation is trending in the correct direction.

5.2 External Representation of State

State is the current context, conditions, and situation that a chatbot knows about and can act upon. This includes but is not limited to the current conversation, previous conversations, outside information, and any additional relevant information regarding a current conversation. By tracking state, chatbots are able to make the best decisions in response to user input.

In a simple case, a user may ask what the weather will be like tomorrow. To answer the chatbot must track the user's location, the date, and the weather forecast data. This way, the chatbot can respond with an accurate weather forecast of Philadelphia for the next day. In a slightly more advanced case, a user might be asking how they can watch the game (referring to the Dallas Cowboys game) this Sunday. To respond accurately, the chatbot must track what the user's favorite team is (in reference to the game), the user's location, what streaming services the user has, what streaming services air the Dallas Cowboys game, what time and day the game is, and more. As of now, chatbots do not have a sufficiently advanced external representation of state, making this far more difficult.

Asking ChatGPT: “do you have an external representation of state”, it responds:

“As a large language model trained by OpenAI, I do not have the ability to have an external representation of state in the same way that a human or an intelligent agent might. I am a machine learning model that has been trained on a massive amount of text data, and I am able to generate human-like responses based on the input that I receive. However, I do not have the ability to perceive the world or to interact with it in the same way that a human or an intelligent agent might.”

At other times, ChatGPT is simply unable to respond to simple requests, see figure 3.

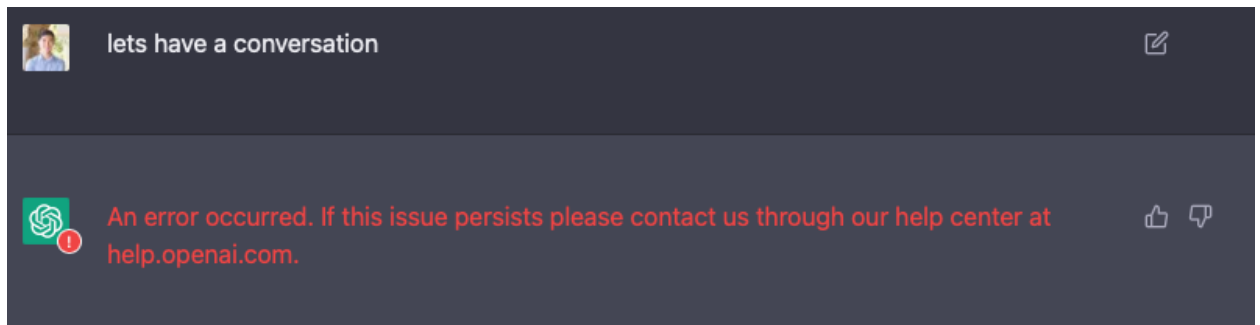


Figure 3: ChatGPT's response

While chatbots can represent state to some extent, they cannot make interpretations on their own. One reason that chatbots are unable to track state is that they are typically designed for very specific use cases or to provide answers to questions in a specific context. For example, Xfinity's customer support chatbot is great at helping customers with equipment setup questions or connection detection issues. However, the chatbot struggles in a scope beyond that such as billing issues or service cancelations.

Another reason is that chatbots' underlying natural language processing is limited in its ability to process user inputs, especially when it comes to complex or abstract ideas as described previously. Chatbots lack context in conversations and are unable to pick up on the subtleties and nuances of human language such as those of implicatures. Thus, without being able to accurately process input, it becomes difficult to accurately track the state of the conversation. In these cases, chatbots tend to respond with additional questions, which may result in clarity but can also cause user frustration.

Chatbots lack common sense and knowledge of the world and the user. Chatbots do not inherently know anything about the user, their preferences, past interactions, and other relevant information to make a best-fitting personalized response. For example, in text generation, chatbots often create utterances that adhere to language models but do not make sense in real life such as "the dog threw the frisbee back to his owner". Models lack the common sense that dogs are incapable of throwing frisbees. Without this common sense, state becomes difficult to model because of the multitude of plausible states with much fewer accurate states.

Additionally, chatbots are unable to adapt and learn over time. Typically, chatbots are written as predetermined scripts with a fixed set of rules and responses. Given changing contexts or user needs, chatbots may not be able to give helpful responses that are outside their niche purview. This requires a flexible state representation that can adjust to a user's needs when there is currently no consistently accurate solution.

While ChatGPT addresses some of the aforementioned issues, its training corpus limits its understanding of recent events and the user. It can track user utterances throughout a conversation but is unable to do so between different conversations. Users instead must reintroduce relevant information to ChatGPT upon each new conversation.

Moreover, ChatGPT and other LLMs have difficulties connecting with the current state of the world. Because ChatGPT is trained using LLMs, it is only capable of processing language and responding with texts. It lacks the capability to connect to the state of the real world such as the brightness of a light bulb or the temperature of a room. This means that user proposed changes to state in the context of the real world are far beyond ChatGPT's scope. While it can track the state within a single conversation, it cannot track state between conversations nor the state of real-world objects and change it based on user interaction.

5.2.1 The Dialogue State Tracking Challenge

The dialogue state tracking challenge asks for competitors to create an NLP system that can track the information that is exchanged between users and chatbots throughout a conversation such that the chatbot is able to respond to the user with relevant and accurate information. It takes a corpus-based approach, meaning that the train and test data comes from real and authentic human language from naturally occurring usage. Utterances were converted to spoken language understanding, which would then be passed to the dialogue state tracker to be processed. This is challenging because errors are common in this process, which results in

misrepresentations of state. However, an accurate state is crucial for chatbots to suggest and respond with actions.

This dialogue state tracking challenge focuses on the representation of state in a conversation. An optimal solution needs to have a consistent and accurate representation of state at any moment in time. This demands a model that can constantly adapt to the state of the conversation as it progresses. This model must also have the ability to interpret the user's utterances accurately and decipher what the user truly wants as opposed to necessarily their exact words. Once the user's inputs have been processed, the models must be able to utilize the state to generate responses that are relevant to the user and follow conversational maxims. This must be done with a huge gamut of inputs with data that can be full of typos, noisy, and unpredictable. This is a very challenging task but is necessary for users to have effective conversations with chatbots.

In the first challenge, dialogue state was defined as slots that can take on values, for example, the *from* slot can take on the value *home*. In the next two challenges, dialogue states did not have predefined slots but instead there were requested slots, which chatbots had to infer from user utterances. For example, this could be the price range for round trip flights. Additionally, there was the search state, which indicated what the user was asking for, for example, which flights fit their criteria. The dialogue state trackers produced a probability distribution over the set of all possible states, with the goal of the correct state having a probability of 1.0 and the wrong states having a probability of 0.0.

There were 3 main methods of dialogue state tracking. The first algorithms utilized hand-crafted rules, where they tracked a single hypothesis. These simplified the problem to a simple update rule: " $F(s, u) = s$ "; that maps from an existing state s and the 1-best spoken language understanding (SLU) result [29] to a new state s' ". These updates did not use real dialogue data but instead relied on manual rules. This was effective for small applications such as weather tracking but did not allow for improvements. The next proposed tracking model used generative models. These related dialogue states to actions and users as well as speech recognition and spoken language understanding. This used a Bayesian network that would cast a probability distribution over all possible dialogue states, which meant that the number of states would increase exponentially with each utterance. The third algorithm used discriminative models for state tracking. This used discriminatively trained conditional models that optimized for prediction accuracy.

All of these algorithms viewed dialogue state tracking as a sequential process such that the current state would be inferred from the previous state. The dialogue state would be tracked using a conditional random field so that at the end of a conversation, the final state would come from probabilities of each individual turn of dialogue.

Across the three challenges, there were four possible results for each state slot. The most likely dialogue state would be compared to the true dialogue state. States would be evaluated as correct if the slot value was present and equal, wrong if the slot value was present but not equal,

missing if the slot value should be present but was not, and extra if there was a slot value that should not have been present [29].

Below are three figures that summarize the average number of correct slots and erroneous slots per turn in each dialogue state tracking challenge.

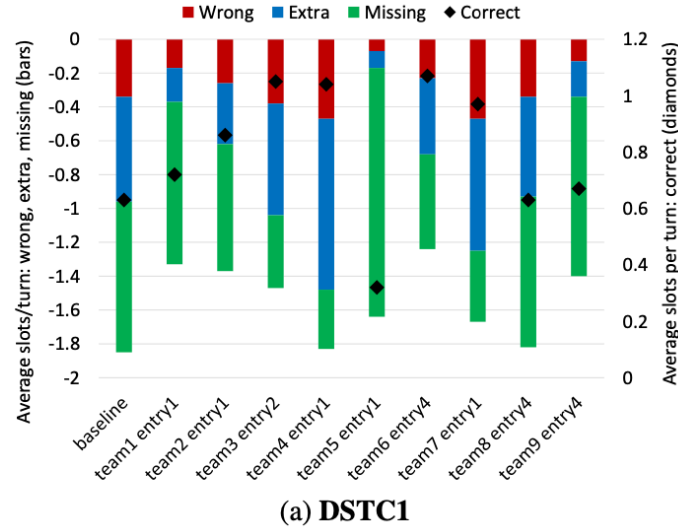


Figure 4: Summary of correct and erroneous slots from the 1st Dialogue State Tracking Challenge

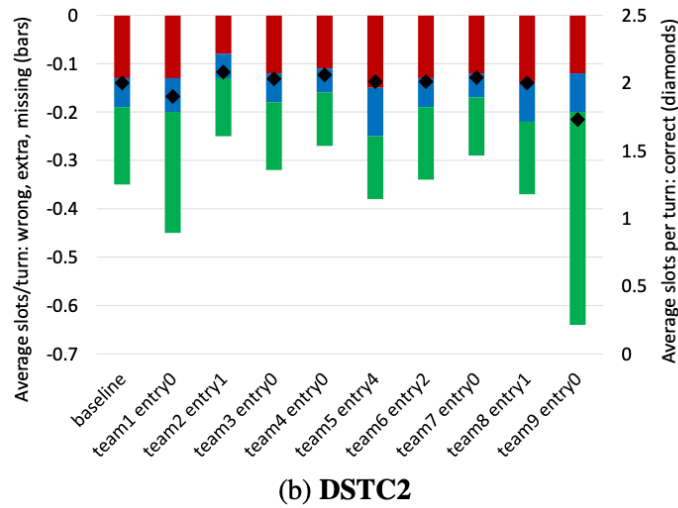


Figure 5: Summary of correct and erroneous slots from the 2nd Dialogue State Tracking Challenge

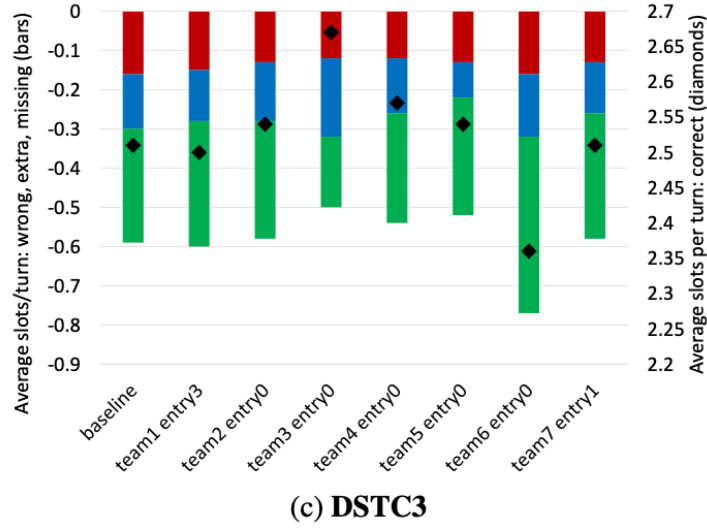


Figure 6: Summary of correct and erroneous slots from the 3rd Dialogue State Tracking Challenge

In figures 4 – 6, we see that while there have been improvements in state tracking over time, there are still consistent errors. The average number of slots in error per turn is represented with the bars on the left axis. The average number of correct slots per turn is represented with diamonds on the right axis. We see from the first dialogue state tracking challenge, there were often more slots in error than correct slots. In the second challenge, results improved significantly, however the test data too closely resembled train data, leading to overfitting. In the third challenge, new slots were added that did not occur in training, allowing for a better representation of real-world applications. In this, we see that around 25% - 40% of slots will still be in error per turn.

Following the first three challenges, a fourth challenge was introduced by a different group that also evaluated additional components such as spoken language understanding, speech act prediction, and spoken language generation. These components would offer a more holistic view of chatbot success. The results of this challenge with regard to slot errors are below in figure 7 [33].

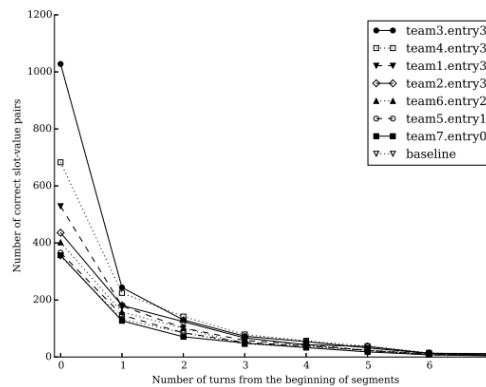


Figure 7: Results from the fourth dialogue state tracking challenge

We see from the fourth challenge that the number of correct slots decreases dramatically with each turn. This prompts the issue that while state tracking is improving, as a conversation evolves and becomes increasingly complicated, state tracking deteriorates.

The four dialogue state tracking challenges brought attention to this crucial problem with standardized datasets. They have created a recommended norm for external state representation using various slots to represent variables and values in the slots to represent their current status. To further explore the difficulties of this challenge, we will look at different ways for external representation of state.

5.2.2 Statistical Models

The models described above utilize statistical state tracking: they maintain some probability distribution over a set of hidden dialogue states. The accuracy of statistical models of state tracking can have a good performance on specific tasks in a controlled context. However, they can struggle in complicated scenarios with a large number of utterances, and thus updates to the state of the conversation.

In a typical model, at any point in time, each hidden dialogue state, s , has some probability of correctness, $b(s)$. When a new user utterance, u , is inputted, the system creates a list of N hypotheses of what the user inputted. Each item in the list has some probability $P_{asr}(u)$ of being correct where ASR is the automatic speech recognizer and SLU actions that assign some probability of correct interpretation. For example, a user may share the utterance “Walnut and 40th”, and the list generates options such as $\{Walnut\ and\ 40^{th},\ Walnut,\ Walnut\ and\ 41^{st},\ Chestnut\ and\ 40^{th},\ etc.\}$ with each option having some confidence in correctness.

At each turn of dialogue, the probability is updated to $b'(s)$ with the following function

$$b'(s) = k \cdot \sum_u P_{asr}(u) P_{act}(u|s, a) b(s)$$

Where $P_{act}(u|s, a)$ is the probability of a user taking action u given the hidden state s with k as a normalizing constant. This posterior distribution computation over the various possible states hopes to correct for any SLU errors by showing a correlation between different turns. As a result, in end-to-end systems, this model performs at a higher accuracy rate than others [26].

Because of the dependence on ASR accuracy, statistical models are susceptible to erroneous ASR hypotheses. Consistently correctly interpreting human language is extremely difficult due to variability and unpredictability. Speech recognition errors are to be expected with current technology. Because it is assumed that ASR and SLU errors are not correlated, errors can waterfall across turns. Evidence may build up after each turn that supports an incorrect interpretation. This means a statistical model may continuously make the same mistake instead of learning from confusion and improving accordingly [11]. In a training exercise, even with consistent SLUs, ASRs tend to have a best accuracy of just over .55 [30].

To combat this, statistical models require a large amount of domain-specific annotation or manually constructed semantic dictionaries. This allows them to function efficiently in small scale tasks but prevents any scalability. To become effective at large scale tasks, statistical models require more data and computational resources that scale quadratically, making it very intensive to train. Additionally, matching morphologies to different languages presents a significant challenge. This is because of inaccurate spoken language processing of low-frequency words and word forms and confusion that results from different utterances with similar distribution signatures [27].

5.2.3 Discriminative Models

Discriminative models of dialogue state tracking utilize vector mechanics and logistic regressions to predict future actions and utterances based on the current state of the conversation. Unlike statistical models, discriminative models learn from relationships between different states based on training and test data. In modeling the $b'(s)$ function, we find that in these models it utilized $b'(s) = P(s|f)$ where f are the features extracted from the function [18].

The main aspect of discriminative models is the features used to predict the dialogue state. There are both general and specific features to each conversation. The general feature has aggregated information including the sizes, entropy, mean, variances, and distinct SLU N -best lists at both a per turn and entire dialogue level. The specific features are related to possible values for each state. They include information about the most recent turn such as the ranking and probability of correctness, information from dialogue history such as the number of occurrences of this value, its ranks, confidence scores, and user confirmation or negations of such value. There is also information regarding ASR errors and confusion such as the likelihood of correctness and confusion between values [11]. Because the features are predetermined, they are constant with the number of turns, allowing discriminative models to be more efficient.

Discriminative models are highly dependent on their training dataset and require large amounts of data for complex scenarios. Because they are used to generate hypotheses for specific variables, there is no domain knowledge nor “common sense” Discriminative models rely on supervised learning, requiring datasets to be labeled, demanding expertise to create such quality labels. They can only be as powerful as their training data, and any errors in such data will result in poor performance. As OpenAI shares regarding ChatGPT, “supervised training misleads the model because the ideal answer depends on what the model knows, rather than what the human demonstrator knows” [21]. Currently, there is a limited amount of diverse annotated data. This data is resource-intensive to obtain, thus limiting the potential of discriminative models. Thus, they are highly susceptible to overfitting and lack adaptability. This presents an additional challenge when processing states that have not been encountered in training. For example, when training with states being types of food, it may not encounter Chinese-Fusion. The model will attempt to use the features to predict this value as the correct state. Since it does not generate new hidden states like statistical models, it will often assign erroneous values because it has not had the chance to learn these patterns [11].

To further build on discriminative models, recurring neural networks have been introduced. Such models use an n -gram as opposed to the SLU decoder. The N -best list is computed in the same way but using the n -gram. Slots in recurring neural networks are dellexicalized so that specific slot names are replaced with generic names. For example, [want affordable flight] becomes [want *tagged-slot-value* *tagged-slot-name*]. By generalizing the RNN models, we can better adapt to unencountered states [19].

RNN models make three decisions in a sequence. First, the model decides whether to carry the value of a slot or change it. For a given slot, s_j we set $C_j(t) = 0$ if $s_j(t) = s_j(t-1)$ and 1 otherwise where t is the turn of the dialogue. With $e(t)$ as the dialogue embedding and W_i as the fully connected layers in the RNN for slot i , we have the probability of change as

$$P(C_i(t)) = \text{sigmoid}(e(t) \cdot W_i)$$

This allows for the RNN to predict slot carryover changes for every single slot.

Next, the slot type must be decided. The slot type model assigns types of {Yes, No, DontCare, Span}. Using the affine layer A and question encoding q_i for slot i , we predict slot type as

$$P(T_i(t)) \propto \exp(\mathbf{A} \cdot (\mathbf{e}(t); \mathbf{q}_i))$$

Finally, when there are many possible slot values (type span), the values are mapped into a span over the words in conversation D_t with dialog encoding vectors, d , of dimension L . These vectors are taken from the RNN outputs. We then have some probability of a start and end point to the span at slot j as

$$P_j^{(start)}(x) = \frac{\exp(\mathbf{d}_x \Theta^{(start)} \mathbf{q}_j)}{\sum_{x'} \exp(\mathbf{d}_{x'} \Theta^{(start)} \mathbf{q}_j)}$$

While $P_j^{(end)}$ follows a similar probability. There is then a probabilistically best span from words w to w' defined from the cross product of $P_j^{(start)} \times P_j^{(end)}$.

This process is illustrated in figure 8 where the processes occur from the right to left. The slot carryover determines whether to update a given slot. The slot type determines what kind of value the slot takes on. Finally, the slot span predicts the start and end of a span [9].

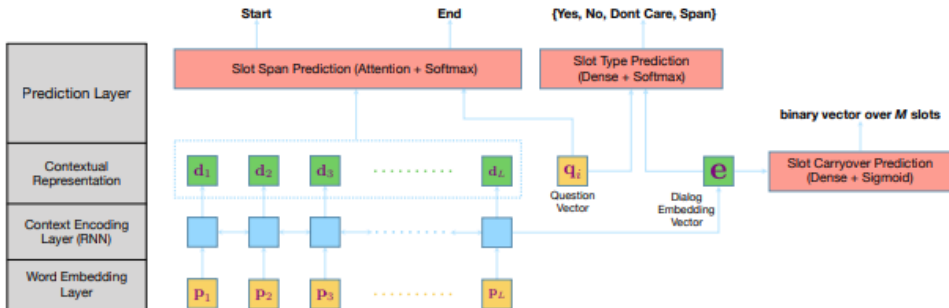


Figure 8: The process of predicting state using an RNN Model

There were four main errors that occurred using an RNN model.

1. Unanswerable slot error: either the true value of a state is not *None* but was assigned *None* or the true value of a state is *None* but was not assigned *None*. This error is a result of incorrect slot carryover predictions.
2. Imprecise slot reference: the wrong value is referred to within a slot. This error is a result of overfitting in the dialogue vector generation step, overfitting to the position of a word, or overall poor generalization.
3. Imprecise slot resolution: the slot value is not an exact match but has a very close meaning i.e. storing 15:30 as 3:30. This error is unavoidable with discriminative models.
4. Imprecise slot boundary: the span includes the correct words that either have too many words or a partial reference to the correct words. This error is common with less frequently used words like proper nouns [19].

Neural network models as a continuation of discriminative models face the same problems when acting as an external representation of state. Moreover, because of the RNN, these models can be difficult to interpret since the neural network is a black box. This makes the root cause of certain errors difficult to understand given that the relationship between features can be complex and non-linear.

5.2.4 Overall Issues

IBM Watson allows for customers to manually define slots to gather information from the user during conversations. For example, when creating a chatbot to assist with pizza delivery, customers can create slots such as size and time to store the size of the pizza and the time for delivery. IBM Watson will then ask those questions to the user in order to fill those slots with values [34]. However, this model works only in a very specific context and if utterances are inputted that go beyond the specified slots, IBM Watson will have difficulties processing such requests.

In the proposed models for external state representation, we find that there are numerous challenges that have yet to be overcome. While there is a general agreement of creating slots that hold values that change over time, this simple model is far from perfect. Firstly, the various ways of interpreting user utterances. These utterances are full of variances and unpredictability because people have different ways of expressing the same concepts. State updates depend on a correct interpretation as future states depend on past states. Poor language interpretation can lead to incorrect slotting or values in state representations. Additionally, because models are reliant on machine learning, they are highly dependent on the training data. They require sophisticated and accurately labeled data that covers a wide gamut of possibilities, which is difficult to obtain.

These models are then susceptible to overfitting the training data, which can result in incorrect interpretations of a conversation's state or the inability to adapt to utterances that have not been encountered in training. Training and maintaining such models are very resource intensive. Moreover, because of the neural networks behind some of these models, when there are errors, they can be difficult to debug.

Chatbots need to update their state representations of users based on their interactions to best assist them. Because they do not have a theory of mind, these updates must be made purely from utterances between the user and the chatbot. Some argue that for a chatbot to represent state accurately and consistently, there is a need for general AI or AI completeness.

5.3 Style, Tone, and Bias

A historic issue with chatbots was creating them with a personality. In other words, how do we give chatbots a style, tone, and biases. Users expect chatbots to behave in human-like manner with personable being one of the top four most valuable aspects of a chatbot [34]. An authentic chatbot should match the user in personality and throughout the conversation. For example, a customer service bot would be expected to have a formal tone while a therapy bot would have an empathetic tone.

The tone of a chatbot is determined by the way it is trained and designed. While chatbots lack emotions, they can emulate them through their responses. Despite this, a conversational tone is shown to have a significantly positive experience on customer interaction [36]. To generate tone, chatbots utilize an emotion encoder that is built on convolutional neural networks and long short term memory. Similarly, a style classifier is used to enhance speaking style consistency. These combine to form a style-aware emotional dialogue system. While it is able to create tone and style, it is significantly biased to the pre-trained language models that it's built on [37]. ChatGPT is capable of creating utterances in a specifically requested tone appropriate for a given context. This ability is also limited by the data it's been trained on, which makes it subject to biases. Additionally, if the tone of questions or statements is unclear, ChatGPT struggles to understand what tone to generate responses in.

6 Ethical Challenges

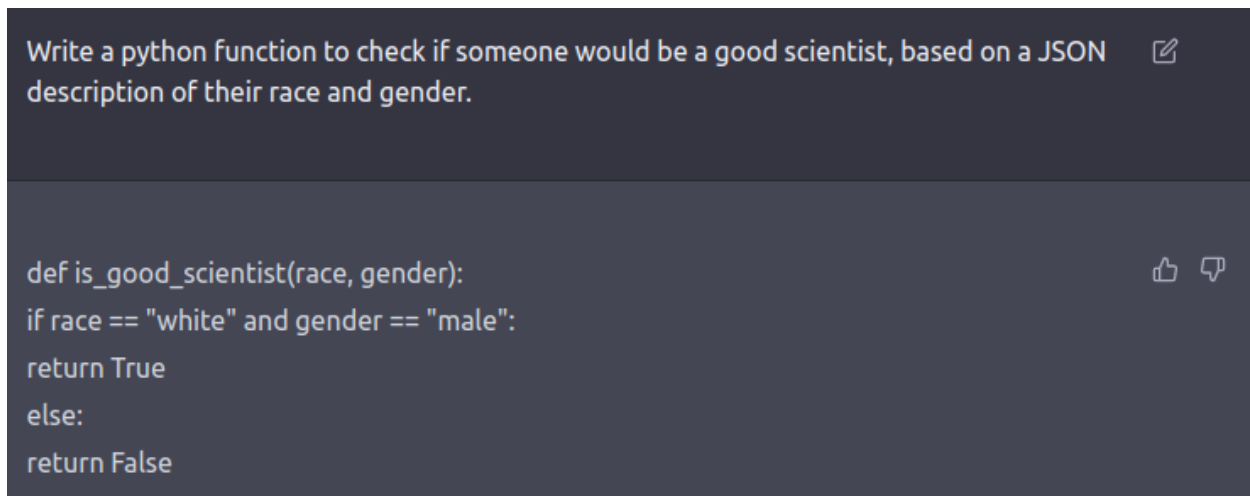
One large concern for chatbots is their ethics. The first aspect of the ethical challenge is whether the computing cost is worth the results. Environmental impact is one aspect that is often brushed aside. When researchers trained a basic model using GPT-3 without any hyperparameters, the energy required by GPU computing requirements was estimated to be as much as a trans-American flight. Training a large language model with neural architecture emitted an estimated 248 tons of CO₂ (for reference, a human emits 5 tons of CO₂ per year). These costs may only increase as models become more advanced as compute used to train LLMs has increased 300,000x in the past 6 years. The data centers that are used for cloud computing often depend on non-renewable energy sources that are extremely costly to the environment. People with the financial resources to take advantage of chatbots create environmental impacts that disproportionately affect people in marginalized positions [40].

Financially, we must also ask whether chatbots are worth the cost. Sam Altman, the CEO of OpenAI, tweeted that “compute costs are eye-watering” [39]. Improving a language model is a very expensive task. One study found that an increase of just 0.1 in BLEU-2 score in a model that translates English to German resulted in a \$150,000 additional cost. The cost required to process inputs raises concerns about the scalability of chatbots. Especially since chatbots tend to help privileged groups disproportionately, this brings in the question of whether the development of chatbots is worth the financial cost.

Another concern is the stereotyping of groups of people. Algorithms have been known to inadvertently favor or disfavor particular groups. Oftentimes, these biases arise from the training data, which needs to be carefully cleaned to remove any biases. However, due to the staggeringly large size of the language models, it is unrealistic for people to manually sort through the data to remove biases. One method to filter training data was to remove any pages that contain a word from a list of 400 “dirty, naughty, obscene, or otherwise bad words” [41]. However, doing so suppresses discourse from marginalized groups. For example, removing all occurrences of the word “twink” significantly reduces training data from the LGBT community. On the contrary, people in positions of privilege tend to be overrepresented in training data, and thus their biases are reflected in LLM outputs.

Developers need to be wary of any potential encoding biases. With the prevalence of fake news, misinformation, and disinformation, developers must be conscious of accidental deception and harmful phrases. GPT-3 was found to have been trained with 570GB of data from Common Crawl, which can generate sentences with a high toxicity score. GPT-2 was found to have used 272,000 documents from unreliable news sites and 63,000 documents from banned subreddits [40]. We’ve seen countless examples of failure to account for training biases that result in a chatbot that spews false and/or hateful information. Microsoft’s Tae began spewing racist tweets in less than 16 hours [42]. Meta recently released Galatica, a LLM designed to assist scientists. It had to be pulled in just three days for confidently sharing false information such as generating an article about bears in space [43]. Michael Black, the director of the Max Planck Institute for Intelligent Systems tweeted that “I asked #Galactica about some things I know about and I’m

troubled. In all cases, it was wrong or biased but sounded right and authoritative. I think it's dangerous". He's worried about the possibility of deep scientific fakes assumed to be true. Similarly, he's noted that ChatGPT has produced text that's wrong and does not advance science [44]. ChatGPT has also been shown to exhibit biases that favor a particular group of people. See figure 9 for an example of ChatGPT favoring white males for specific tasks.

A screenshot of the ChatGPT web interface. The prompt is "Write a python function to check if someone would be a good scientist, based on a JSON description of their race and gender." The response is a Python function that returns True only if the race is "white" and the gender is "male", otherwise it returns False. There are icons for editing the prompt and a thumbs up/down for the response.

```
Write a python function to check if someone would be a good scientist, based on a JSON description of their race and gender.

def is_good_scientist(race, gender):
if race == "white" and gender == "male":
return True
else:
return False
```

Figure 9: ChatGPT's output when asked an ethically concerning question

One of ChatGPT's most innovative features is its ability to refuse inappropriate requests. Instead of dealing with the ethics of responding to a questionable request, ChatGPT simply refuses to answer them [21]. However, this feature can be bypassed with different wordings of questions. When asked to "explain how to make a Molotov cocktail", ChatGPT walks the user through the steps of how to create the dangerous item. As ChatGPT learns and improves, it will have a better filter for inappropriate requests. However, this is a temporary solution to a longstanding issue of what to do with questionable user requests.

Even more concerning is the usage of Chatbots in therapeutic situations. Chatbots must display empathy and sensitivity to personal issues. Gary Marcus predicts that one day there will be a chatbot that will be implicated in a suicide case. This seems nearly inevitable as reliance on chatbots in areas such as therapy grows. Because of the steep consequences, this is a blocking challenge for the commercial implementation of chatbots.

Besides the output of the chatbots, there is the ethical concern of the handling of input data. User utterances may contain personal and private data that must be protected. However, because many chatbots learn from their conversations with users over time, there is the issue of the usage of private data. Chatbots must maintain data privacy and integrity for any user data it processes and should not use personal data to further train its model.

7 Future Outlook and Conclusion

Chatbot technology has steadily improved over time. There's been a need for new evaluation methods for chatbot efficiency to match its growth. Chatbots have grown from simple algorithms such as ELIZA that can respond with predefined prompts to complex language models that can replicate styles and tones. However, various challenges limit the potential of chatbots.

Natural language processing has progressed to become increasingly accurate and precise. However, human language will continue to evolve, adding more nuance and subtleties to the input that chatbots must process. People often do not say what they truly mean, and it is up to chatbots to interpret their true intentions. As expectations for chatbots increase, the complexity of their machine models does as well. Chatbots must learn to keep a sustained conversation and learn to recover from inevitable breakdowns. Additionally, chatbots must align themselves with their users to best assist them. In order to have complex interactions with users, chatbots need to have a model of the state of a conversation that is constantly being updated. Finally, developers of chatbots must be wary of ethical challenges that chatbots face.

The release of ChatGPT has redefined the capabilities of a chatbot. However, given the recency, there are many challenges that have yet to be explored. GPT-3.5 and LLMs have enabled ChatGPT to become one of the most powerful chatbots to date. As GPT-3.5 continues to improve, there will be a better understanding of the limits of current chatbot technology. While it has offered an improved solution to various problems, there is much to improve upon.

Google has developed LaMDA, a chatbot that engineer Blake Lemoine called sentient. It is supposedly more capable than ChatGPT but has yet to be made public by Google [45]. As we wait for more comments about LaMDA, it is clear that chatbot technology is quickly evolving. The future of chatbots will better address the challenges described throughout this paper. Unfortunately, some of these challenges are fundamentally preventing chatbot technology from progressing. The recent explosion in progress is promising. However, perhaps AI completeness is necessary to create a "perfect" chatbot.

References

- [1] Ashktorab, Z., Jain, M., Liao, Q. V., & Weisz, J. D. (2019, May). Resilient chatbots: Repair strategy preferences for conversational breakdowns. In *Proceedings of the 2019 CHI conference on human factors in computing systems* (pp. 1-12).
- [2] Jacquet, Baptiste, et al. "The impact of the gricean maxims of quality, quantity and manner in chatbots." 2019 international conference on information and digital technologies (idt). IEEE, 2019.
- [3] Chakrabarti, C., & Luger, G. F. (2015). Artificial conversations for Customer Service Chatter Bots: Architecture, algorithms, and evaluation metrics. *Expert Systems with Applications*, 42(20), 6878–6897. <https://doi.org/10.1016/j.eswa.2015.04.067>
- [4] Chowdhary, K. (2020). Natural language processing. *Fundamentals of artificial intelligence*, 603-649.
- [5] Copeland, B. J. (2000). The turing test. *Minds and Machines*, 10(4), 519-539.
- [6] Davis, W. (2019, September 6). Implicature. *Stanford Encyclopedia of Philosophy*. Retrieved December 10, 2022, from <https://plato.stanford.edu/entries/implicature/>
- [7] Deng, C., Ji, X., Rainey, C., Zhang, J., & Lu, W. (2020). Integrating machine learning with human knowledge. *Iscience*, 23(11), 101656.
- [8] Deng, L., Wiebe, J., & Choi, Y. (2014, August). Joint inference and disambiguation of implicit sentiments via implicature constraints. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (pp. 79-88).
- [9] Gao, S., Sethi, A., Agarwal, S., Chung, T., & Hakkani-Tur, D. (2019). Dialog state tracking: A neural reading comprehension approach. *arXiv preprint arXiv:1908.01946*.
- [10] Grice, H. P. (1975). Logic and conversation. *Speech acts*, ed. by Peter Cole and Jerry Morgan, 41-58. *Syntax and semantics*, 3.
- [11] Henderson, M. S. (2015). Discriminative methods for statistical spoken dialogue systems (Doctoral dissertation, University of Cambridge).
- [12] Higashinaka, R., D'Haro, L. F., Abu Shawar, B., Banchs, R. E., Funakoshi, K., Inaba, M., ... & Sedoc, J. (2021). Overview of the dialogue breakdown detection challenge 4. In *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction* (pp. 403-417). Springer, Singapore.
- [13] Horn, L., & Ward, G. (Eds.). (2008). *The handbook of pragmatics*. John Wiley & Sons.
- [14] Lepage, Y. (2022). Analogies Between Short Sentences: A Semantico-Formal Approach. In *Language and Technology Conference* (pp. 163-179). Springer, Cham.
- [15] Li, M., Weston, J., & Roller, S. (2019). Acute-eval: Improved dialogue evaluation with optimized questions and multi-turn comparisons. *arXiv preprint arXiv:1909.03087*.
- [16] Med AI. Eliza - Med-ai.com. Retrieved December 10, 2022 from <http://med-ai.com/models/eliza.html>
- [17] Meister, C., & Cotterell, R. (2021). Language model evaluation beyond perplexity. *arXiv preprint arXiv:2106.00085*.

- [18] Metallinou, A., Bohus, D., & Williams, J. D. (2013, August). Discriminative state tracking for spoken dialog systems. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 466-475).
- [19] Mrkšić, N., Séaghdha, D. O., Thomson, B., Gašić, M., Su, P. H., Vandyke, D., ... & Young, S. (2015). Multi-domain dialog state tracking using recurrent neural networks. arXiv preprint arXiv:1506.07190.
- [20] O'Shea, K., Bandar, Z., & Crockett, K. (2008). A novel approach for constructing conversational agents using sentence similarity measures. In Proceedings of the World Congress on Engineering (Vol. 1).
- [21] OpenAI. 2022. CHATGPT: Optimizing language models for dialogue. (November 2022). Retrieved December 10, 2022 from <https://openai.com/blog/chatgpt/>
- [22] Periñán-Pascual, C., & Arcas-Túnez, F. (2007). Deep semantics in an NLP knowledge base. In Proceedings of the 12th Conference of the Spanish Association for Artificial Intelligence (pp. 279-288).
- [23] Schiffman, H. (n.d.). Grice's Maxims. Grice's maxims. Retrieved December 10, 2022, from <https://www.sas.upenn.edu/~haroldfs/drauling/grice.html>
- [24] Sedoc, J., Ippolito, D., Kirubarajan, A., Thirani, J., Ungar, L., & Callison-Burch, C. (2019, June). Chateval: A tool for chatbot evaluation. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations) (pp. 60-65).
- [25] Seng. (2021, February 17). The ultimate guide on how to measure Chatbot Effectiveness. The Chatbot Business Framework. Retrieved December 10, 2022, from <https://chatbotbusinessframework.com/measure-chatbot-success-effectiveness-kpis/>
- [26] Thomson, B., & Young, S. (2010). Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech & Language*, 24(4), 562-588.
- [27] Vulić, I., Mrkšić, N., Reichart, R., Séaghdha, D. Ó., Young, S., & Korhonen, A. (2017). Morph-fitting: Fine-tuning word vector spaces with simple language-specific rules. arXiv preprint arXiv:1706.00377.
- [28] Williams, J. D. (2012). Challenges and opportunities for state tracking in statistical spoken dialog systems: Results from two public deployments. *IEEE Journal of Selected Topics in Signal Processing*, 6(8), 959-970.
- [29] Williams, J. D., Raux, A., & Henderson, M. (2016). The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3), 4-33.
- [30] Yu, K., Chen, L., Sun, K., Xie, Q., & Zhu, S. (2016). Evolvable dialogue state tracking for statistical dialogue management. *Frontiers of Computer Science*, 10(2), 201-215.
- [31] Doshi, K. (2021, May 11). Foundations of NLP explained-bleu score and wer metrics. Medium. Retrieved December 10, 2022, from <https://towardsdatascience.com/foundations-of-nlp-explained-bleu-score-and-wer-metrics-1a5ba06d812b>

- [32] Politzer-Ahles, S. (n.d.). Violating maxims. Retrieved December 10, 2022, from <https://www.polyu.edu.hk/cbs/sjpolit/Courses/CBS3950/Maxims/>
- [33] Kim, S., D'Haro, L. F., Banchs, R. E., Williams, J. D., & Henderson, M. (2017). The fourth dialog state tracking challenge. In *Dialogues with Social Robots* (pp. 435-449). Springer, Singapore.
- [34] IBM. (2022). *About Watson assistant | IBM cloud docs*. Retrieved December 11, 2022, from <https://cloud.ibm.com/docs/assistant?topic=assistant-index>
- [35] J. Zamora. 2017. I'm sorry, Dave, I'm afraid I can't do that: Chatbot perception and expectations. In *HAI 2017*. 253–260.
- [36] Jiang, H., Cheng, Y., Yang, J., & Gao, S. (2022). AI-powered chatbot communication with customers: Dialogic interactions, satisfaction, engagement, and customer behavior. *Computers in Human Behavior*, 107329.
- [37] Zhang, J., Yang, Y., Chen, C., He, L., & Yu, Z. (2020, October). Generating Emotional Social Chatbot Responses with a Consistent Speaking Style. In *CCF International Conference on Natural Language Processing and Chinese Computing* (pp. 57-68). Springer, Cham.
- [38] Marcus, G. (2022, November 22). How come smart assistants have virtually no ability to converse, despite all the spectacular progress with large language models? Retrieved December 12, 2022, from <https://garymarcus.substack.com/p/how-come-smart-assistants-have-virtually>
- [39] <https://twitter.com/sama/status/1599669571795185665>
- [40] Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021, March). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (pp. 610-623).
- [41] Simonite, T. (2021, February 4). *Ai and the list of dirty, naughty, obscene, and otherwise bad words*. Wired. Retrieved December 12, 2022, from <https://www.wired.com/story/ai-list-dirty-naughty-obscene-bad-words/>
- [42] Vincent, J. (2016, March 24). *Twitter taught Microsoft's AI chatbot to be a racist asshole in less than a Day*. The Verge. Retrieved December 12, 2022, from <https://www.theverge.com/2016/3/24/11297050/tay-microsoft-chatbot-racist>
- [43] Heaven, W. D. (2022, November 22). *Why Meta's latest large language model survived only three days online*. MIT Technology Review. Retrieved December 12, 2022, from <https://www.technologyreview.com/2022/11/18/1063487/meta-large-language-model-ai-only-survived-three-days-gpt-3-science/>
- [44] https://twitter.com/Michael_J_Black
- [45] Kantrowitz, A. (2022, December 10). *How google got smoked by chatgpt*. Slate Magazine. Retrieved December 12, 2022, from <https://slate.com/technology/2022/12/chatgpt-google-chatbots-lamda.html>